

Treball de Fi de Grau

Grau en Enginyeria en Tecnologies Industrials

Sistema de tractament d'imatges per a la mida de nanopartícules

Memòria

Autor: Ferran Calvo López

Director: Jordi Bou Serra

Convocatòria: Gener 2018



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Sumari

SUMARI	2
RESUM	4
1. INTRODUCCIÓ	5
1.1. Origen i motivació del projecte	5
1.2. Abast del projecte	6
2. OBJECTIUS	7
2.1. Objectiu general	7
2.2. Objectius particulars	7
2.2.1 Obtenció de mides de partícules a través del reconeixement dels contorns de les partícules a través de l'ús d'algoritmes	8
2.2.2 Obtenció de mides de partícules per via de traçat de cercles	8
2.2.3 Presentar un mètode informàtic d'exportació dels resultats a programes de tractament de dades tradicionals	9
3. LENGUATGE DE PROGRAMACIÓ: PYTHON 3	10
4. IMATGES DE PARTÍCULES OBTINGUDES PER MICROSCÒPIA ELECTRÒNICA	12
5. LA IMATGE DIGITAL	14
5.1. Tipus	14
5.2. Píxel	15
5.3. Resolució d'una imatge	16
5.4. Profunditat i models de color	16
6. TRACTAMENT D'IMATGES	18
6.1. Realç o millora de la imatge	19
6.1.1. Soroll d'una imatge	20
6.1.2. Soroll gaussià	21
6.1.3. Soroll de sal i pebre	22
6.1.4. Filtrat de la imatge	22
6.1.4.1. Filtres de passa baix	23
6.1.4.2. Filtres de passa alt	24

6.1.4.3. Filtres direccionals	24
6.1.4.4. Filtres de detecció de contorns.....	24
6.1.5. Detecció de contorns, l'algoritme de Canny	25
6.1.5.1. Obtenció del gradient.....	25
6.1.5.2. Supressió no màxima	26
6.1.5.3. Histèrsis de llindar a la supressió no màxima	26
6.1.5.4. Exemple gràfic de l'aplicació de l'algoritme Canny	27
6.2. Aplicació dels algorismes anteriors a la mesura de les nanopartícules	27
7. TRAÇAT DE CERCLES	32
8. RESULTATS	36
8.1. Codi del programa	36
8.2. Esquema general del programa	45
8.3. Algoritme del programa.....	47
8.4. Funcionalitats del programa i guia per l'usuari	49
8.4.1. Obrir imatge	49
8.4.2. Dibuixar escala.....	50
8.4.3. Dibuixar contorn i determinar diàmetre de les partícules.....	53
8.4.4. Numeració de les partícules	55
8.4.5. Generació del fitxer de dades.....	57
8.4.6. Manipulació de les dades	59
8.5. Possibles funcionalitats i millores del programa.....	60
8.5.1. Funció desfer	60
8.5.2. Determinar escala a través del programa	60
9. ESTUDI ECONÒMIC DEL PROJECTE	61
10. IMPACTE AMBIENTAL DEL PROJECTE	63
11. PLANIFICACIÓ	64
CONCLUSIONS	66
AGRAÏMENTS	68
BIBLIOGRAFIA	69

Resum

Actualment, la mesura de nanopartícules captades a través de microscòpia electrònica és molt important per les aplicacions d'aquestes en diferents àmbits com ara la química i biomedicina que és el cas que ha promogut el projecte.

Ara bé, actualment no existeixen gaires eines informàtiques a l'abast per tal de fer aquest procés de mesura senzill, ràpid i precís. Molts cops s'ha d'optar a mesurar-les a mà amb un regle sobre la imatge que conté les nanopartícules i l'escala que hi apareix.

Aquest és el problema que em va presentar el professor Jordi Bou Serra, tutor del projecte, del Departament de Química de l'Escola Tècnica Superior d'Enginyeria Industrial de Barcelona.

En el present projecte, s'han estudiat diverses vies per tal de solucionar aquesta problemàtica. Una d'elles ha estat l'automatització completa d'aquest procés a través d'un programa informàtic, cosa que, com es veurà posteriorment, no ha estat possible.

Finalment, s'ha optat per una solució alternativa intermèdia que requereix d'una certa interacció de l'usuari amb el programa informàtic.

El codi del programa consta aproximadament d'unes 140 línies. El programa, dissenyat amb Python, fa ús de diverses llibreries, incloent les funcions d'obrir una imatge, traçar l'escala i el contorn de les partícules a sobre d'aquesta, crear un arxiu de text amb les dades obtingudes, així com crear una interfície gràfica i menús per accedir a les diferents funcionalitats del programa.

Amb el programa dissenyat únicament amb 3 clics de ratolí es pot dibuixar amb una bona precisió el contorn circular d'una partícula i determinar el seu diàmetre, enfront a l'escassa precisió i excessiva lentitud de l'actual procés manual de mesura.

Tot això ha sigut possible després de fer un estudi exhaustiu de les diferents eines i metodologies de tractament d'imatges que existeixen a l'actualitat i les llibreries disponibles que fan possible aquest tractament.

1. Introducció

1.1. Origen i motivació del projecte

El projecte sorgeix de la necessitat que té el Departament de Química de l'ETSEIB per realitzar mesuraments de nanopartícules esfèriques obtingudes a través de mètodes de captació de imatges microscòpiques per mitjans diverses, com per exemple aparells electrònics, òptics o atòmics.

La nanociència, així com el seu desenvolupament nanotecnològic, s'enfoca principalment al estudi dels components dels materials a una escala atòmica, molecular i macromolecular, per analitzar propietats d'aquests materials que no són observables a escales superiors o per altres mètodes d'observació, amb l'objectiu de comprendre i controlar els fenòmens que es produeixen a aquests nivells i aconseguir manipular aquests fenòmens en interès de la ciència i la tecnologia.

Són múltiples els camps d'aplicació entre els quals poden citar la manipulació de materials, la medicina, la biologia i biotecnologia, l'energia, el medi ambient, les noves tecnologies, l'electrònica, la física i la química entre altres.

Concretament les imatges que presenta el Departament de Química per automatitzar el seu mesurament, que motiven aquesta projecte, són nanopartícules que corresponen a l'àcid polilàctic (PLA), un plàstic biodegradable.

Quantificar les mides d'aquestes nanopartícules és de vital importància per poder controlar les seves aplicacions potencials en els diversos àmbits en els quals s'utilitzen.

Algunes de les principals aplicacions del PLA són:

- Nanoencapsulament de fàrmacs: dins les nanotecnologies trobem les bionanotecnologies que investiguen les interaccions entre els materials nanomètrics i els sistemes biològics.

Els productes de les bionanotecnologies inclouen als nanomedicaments i els seus components substàncies com additius que milloren la biodisponibilitat i solubilitat de fàrmacs que són molt poc solubles, entre d'altres.

Els principals avantatges de l'ús de nanopartícules per l'administració de fàrmacs venen

donades, principalment, per la seva petita grandària i per l'ús de materials biodegradables, el que pot permetre solucionar problemes associats amb la solubilitat, citotoxicitat, biodisponibilitat i immunocompatibilitat dels medicaments tradicionals.

Gràcies a aquesta tan reduïda grandària, les partícules són capaces de travessar, per exemple, tumors o penetrar microcapil·lars. Una altra de les avantatges que presenten les nanopartícules sobre les micropartícules és la seva millor adequació quan s'han d'administrar per via intravenosa, ja que la seva mida és molt més petita que els capil·lars de menor diàmetre del cos humà, el que evita la formació d'agregats i garanteix que les partícules no formin èmbols o trombes.

- Nano i microrecobriments: les partícules formen capes o pel·lícules sobre algunes superfícies de gruixos molt controlats i precisos.

D'aquesta manera s'aconsegueixen unes propietats per la protecció dels agents externs molt controlades, així com també, com ja s'ha citat a l'apartat anterior, es poden fer servir a l'àmbit farmacològic constituint un recobriment biodegradable i/o sostenible.

- Nano i microabrasius biodegradables: com a substituïts preferibles a les partícules de plàstics o de ceràmica no biodegradables.
- Pols per aplicacions en impressió 3D: no han de ser necessàriament nanopartícules, però s'ha de controlar la seva mida segons l'aplicació que se li vulgui donar.

1.2. Abast del projecte

L'abast del projecte és implementar un programa informàtic que permeti millorar el mètode de mesura de les micropartícules i nanopartícules quasiesfèriques obtingudes gràficament pels mitjans esmentats a l'apartat 1.1 Origen i motivació del projecte.

2. Objectius

2.1. Objectiu general

L'objectiu del projecte consisteix en automatitzar el procés de mesura de les partícules que apareixen en les imatges captades per microscòpia i determinar la seva grandària (bàsicament el radi d'aquestes, ja que al tractar-se de partícules quasi esfèriques, que considerarem com a esfèriques, con aquesta dada podrem fer els càlculs de la resta de dimensions de cada nanopartícula.).

Per poder realitzar totes aquestes accions de manera automàtica, dins del camp de la visió artificial, farem un estudi de les diverses eines de tractament d'imatges i reconeixement de patrons que existeixen a nivell de software, amb la finalitat de seleccionar les que millor funcionalitats aportin a la resolució de la qüestió que volem resoldre, o crear-ne una de nova o una combinació de tot dos plantejaments.

D'aquesta manera, en aquest projecte, es pretén trobar una solució pel cas particular que presenta el Departament de Química del centre en quant a la obtenció de les dimensions de les múltiples partícules quasi esfèriques que apareixen a les imatges que volem tractar.

Posteriorment el programa es podria generalitzar o adaptar a altres tipus d'imatges amb la intenció, a més a més, d'oferir la possibilitat de resoldre problemes similars que es presentin en el camps de la química, biologia, medicina, entre d'altres, i que requereixin d'un tractament d'imatges semblant.

2.2. Objectius particulars

Els dos primers objectius particulars que es presenten fan referència a dues possibles vies en les que basar-se a l'hora de dissenyar el programa informàtic i per tal d'arribar a assolir l'objectiu general presentat a l'apartat anterior de la manera més eficient possible.

Com primera opció intentaré realitzar el tractament totalment automatitzat dels mesuraments de les imatges, on la intervenció de l'usuari es limitaria a seleccionar la imatge per obtenir els resultats.

Donat l'estat actual de les eines software i la limitació de temps per al desenvolupament del programa informàtic, si no fos possible encaixar la primera opció, m'inclinaré cap a una solució

que, millorant substancialment el temps necessari i la precisió en les mesures del mètode actual, precisi d'una major intervenció de l'usuari.

2.2.1 Obtenció de mides de partícules a través del reconeixement dels contorns de les partícules a través de l'ús d'algoritmes

El mecanisme de mesura de partícules que es pretén implementar consisteix, en principi, en emprar algoritmes i eines disponibles en diverses llibreries dedicades al tractament d'imatges.

Així doncs, el que es vol aconseguir mitjançant aquest procés de tractament de la imatge és delimitar de manera correcta cadascuna de les partícules que hi apareixen a partir del reconeixement dels contorns de cadascuna per, posteriorment, aproximar cadascun d'aquest contorns a la circumferència que millor s'aproximi al mateix.

Un cop les partícules quasi esfèriques es troben ben delimitades pels seus respectius contorns ja es podrien aproximar a la seva circumferència i procedir a mesurar una a una de manera individual.

Aquest sistema permetria una mesura de les mides de les partícules gairebé de manera automàtica, amb mínima intervenció de l'usuari, i un òptim estalvi de temps ja que la interacció de l'usuari amb el programa és mínima degut a que és el programa el que s'encarrega per si mateix de separar i mesurar les partícules que apareguin en la imatge.

2.2.2 Obtenció de mides de partícules per via de traçat de cercles

En aquest mètode de mesura, a diferència de l'anterior, implica que l'usuari hagi de interactuar de manera considerablement més activa amb el programa la qual cosa, evidentment, incrementarà sensiblement el temps de resposta que ofereix el programa a l'usuari final.

El funcionament del programa en aquest cas es basarà en el traçat de cercles que passaran per tres punts prèviament indicats per l'usuari a la fotografia. D'aquest mode, com les partícules que es volen tractar són pràcticament esfèriques, el cercle traçat s'aproxima d'una manera molt precisa a la mida de la partícula que es vol estudiar.

Una vegada introduïda per part de l'usuari l'escala de la imatge objecte del tractament, que en

funció de la plataforma de eines i llibreries escollida podria ser a l'inici de la execució del programa o al final del mateix directament en la fulla dels mesuraments obtinguts, l'aplicatiu procedirà a la realització dels càlculs matemàtics per calcular la mesura de cadascuna de les partícules seleccionades per l'usuari en la imatge que hagi escollit.

Tot i que aquest procés requereix una certa interacció de l'usuari amb la computadora, la qual cosa augmenta el temps de resposta, presenta algunes avantatges enfront a l'altre sistema més automatitzat, com ara la possibilitat de triar les partícules que són d'interès per a l'usuari (per exemple, si hi ha partícules que no es volen mesurar perquè a simple vista ja es veu que són massa grans, massa petites o que, per tractar-se d'una imatge 2D que es representació d'un espai original real en 3D, per la captació visual per part de l'usuari de la profunditat de la partícula en la imatge aquest decideixi rebutjar-la).

2.2.3 Presentar un mètode informàtic d'exportació dels resultats a programes de tractament de dades tradicionals

Un cop obtinguts els radis o diàmetres de les partícules a través del tractament de la imatge mitjançant el programa informàtic, interessa poder exportar els resultats a altres programes per poder-les tractar i poder fer-ne un estudi més detallat, per aquesta finalitat, en principi, sembla adequat presentar els mesuraments a través de l'Excel.

3. Llenguatge de programació: Python 3

El llenguatge de programació triat per dur a terme l'elaboració del programa informàtic és Python, concretament la versió Python 3.

El motiu principal de la tria d'aquesta opció és que Python és el llenguatge de programació que s'utilitza al llarg de les assignatures de programació del Grau d'Enginyeria en Tecnologies Industrials.

Deixant de banda aquest motiu, Python és un llenguatge que presenta avantatges en diferents aspectes i que resulta una bona opció a considerar si es vol triar un llenguatge per programar.

Alguns d'aquests aspectes són:

- Facilitat d'ús: per una persona que comença a programar o no té gaire experiència en el món de la programació, és un llenguatge que al poc temps permet fer els primers programes senzills.
- Llegibilitat: és un llenguatge que s'escriu d'una forma molt natural i que facilita la lectura a altres programadors si volen fer alguna modificació al codi.
- Compatibilitat: és un llenguatge que es pot usar tant a Mac, com a Windows i a Linux.
- Abundància de llibreries: té una gran biblioteca on es poden trobar moltes llibreries disponibles de Python per estendre la seva funcionalitat a altres camps.
- Gran base d'usuaris: existeix una ampla comunitat que fas ús d'aquest llenguatge, la importància evolutiva del llenguatge (actualitzacions per exemple) i facilitats per resoldre problemes dels usuaris durant la programació.
- Multiparadigma: permet la generació de programes utilitzant més d'un estil de programació: orientada a objectes, imperativa i funcional.
- Resolució dinàmica de variables: disposa d'un sistema de resolució dinàmica de noms, és a dir, un mètode que enllaça el nom d'una variable durant la execució del programa.

- Facilitat d'extensió: té funcionalitats orientades al fet que es puguin escriure nous mòduls en C o C++, per lo que es pot incloure en aplicacions que precisen una interfície programable.
- Immediatesa de l'avaluació: inclou una manera interactiva en el que les instruccions s'escriuen en un intèrpret de comandos que avalua immediatament les expressions introduïdes i prova porcions de codi de forma immediata abans d'integrar-ho en la aplicació.



Figura 1. Llenguatge de programació utilitzat: Python 3

4. Imatges de partícules obtingudes per microscòpia electrònica

A continuació es mostren algunes fotografies d'àcid làctic (PLA) obtingudes per mitjà de microscòpia electrònica i amb diferents rangs d'escales. Aquestes fotografies són les que seran tractades al programa informàtic per dur a terme la mesura de les partícules que hi apareixen.

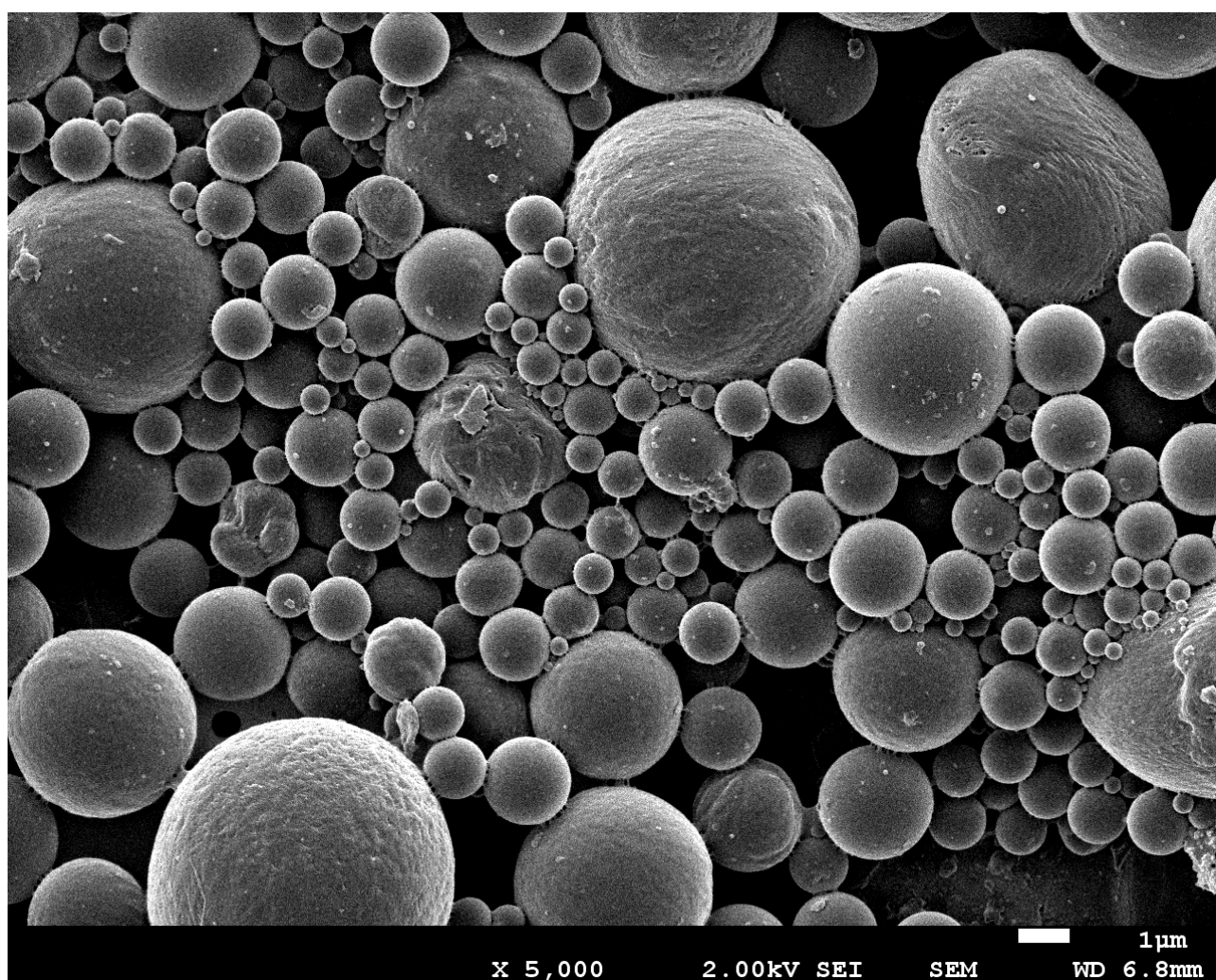


Figura 2. Fotografia de partícules presa amb microscòpia electrònica

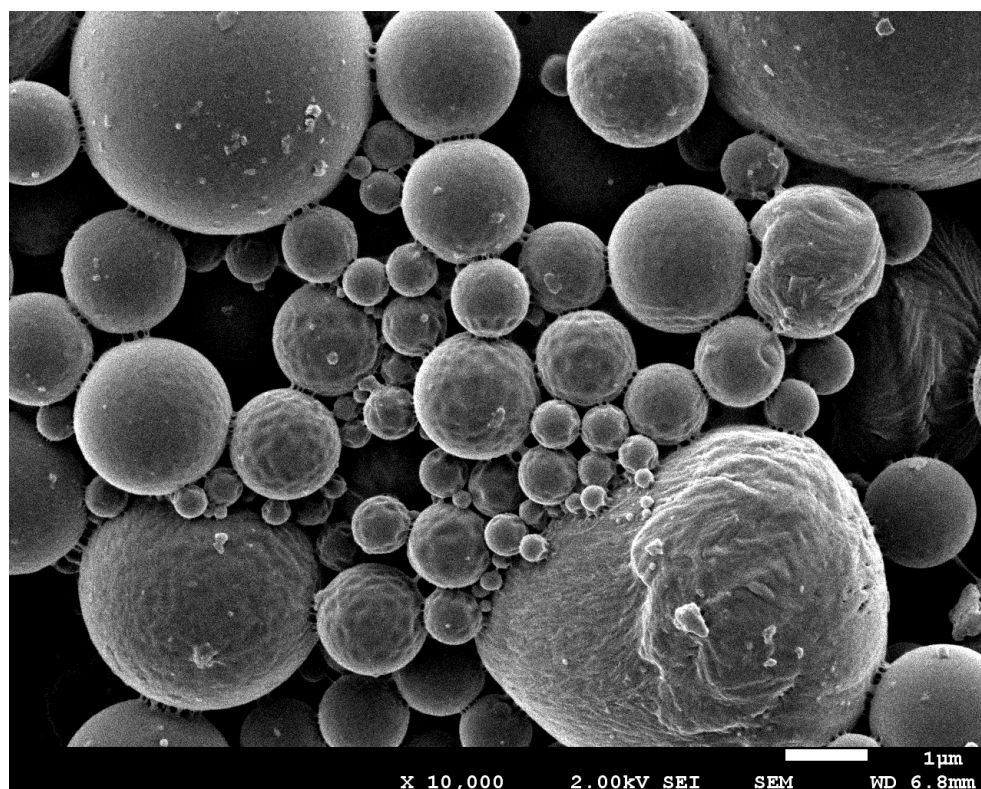


Figura 3. Fotografia de partícules presa amb microscòpia electrònica

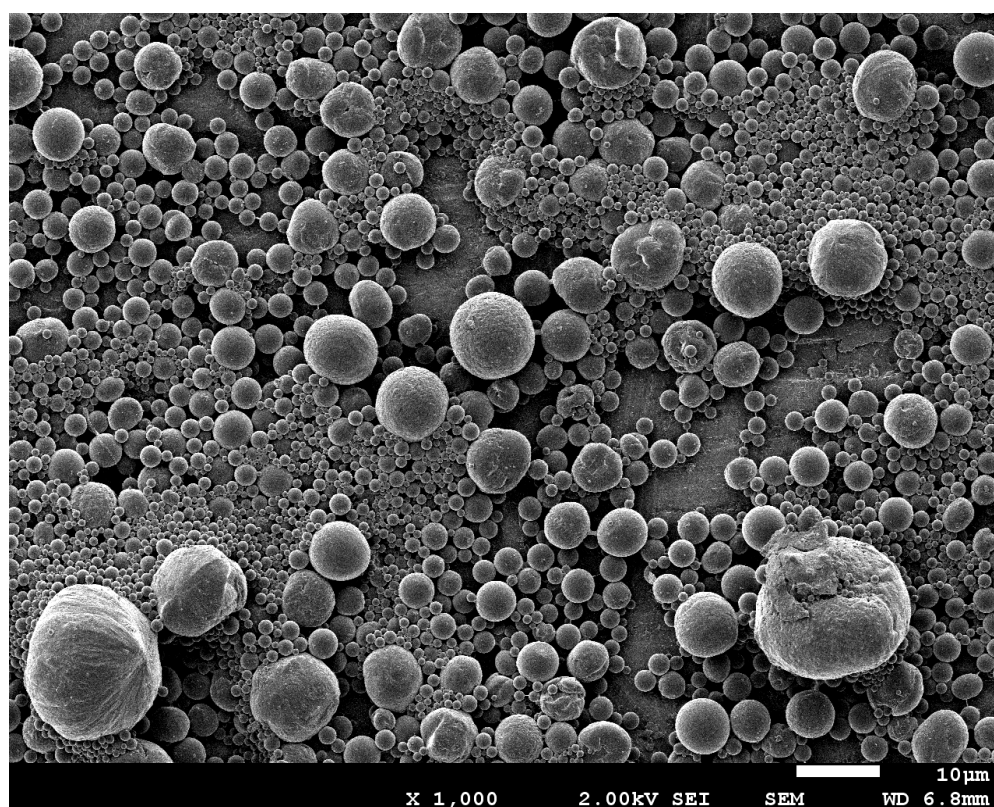


Figura 4. Fotografia de partícules presa amb microscòpia electrònica

5. La imatge digital

La informació que es vol extreure i estudiar (en aquest cas, la mida de les partícules fotografiades) es troba en un fitxer d'imatges de format habitual, com per exemple .jpg, .gif, .png i altres.

Un primer pas, doncs, implica analitzar i entendre com està composta una imatge, els tipus d'imatges que existeixen i els elements principals que les conformen per poder fer un tractament correcte d'aquesta.

5.1. Tipus [1]

Una imatge és la representació visual que manifesta l'aparença visual d'un objecte real o imaginari, en la terminologia informàtica és un arxiu que mostra una presentació visual, com per exemple una foto o un dibuix.

Una imatge digital és la representació bidimensional d'una imatge fent servir bits, unitat mínima d'informació composta per dígit binaris (1 i 0).

Es distingeixen dos tipus d'imatges digitals:

- Imatges vectorials: són aquelles que estan formades per components geomètrics independents.

Aquests components geomètrics poden ser segments o polígons i estan definits per atributs matemàtics que indiquen diverses característiques com per exemple la seva posició o el seu color.

A diferència de les imatges digitals de mapa de bits (que es veuran a continuació), les imatges vectorials, per la seva naturalesa, al estar formades per objectes geomètrics es poden ampliar sense perdre qualitat.

En quant als formats actualment més utilitzats per aquest tipus d'imatges, destaquen el PDF, SVG i VML.

Tanmateix, els editors més utilitzats són Corel Draw, Freehand i Adobe Illustrator.

- Imatges de mapa de bits: també anomenades *raster images*, són imatges que estan compostes per una sèrie de punts, anomenats píxels.

Cada imatge conté un número fix de píxels, cadascun dels quals està ubicat en un punt en concret de la imatge i posseeix uns valors propis que determinen el seu color.

El píxel és per tant, la unitat d'informació bàsica dins la imatge.

Al contrari de les imatges digitals vectorials, en ampliar-se es perd qualitat en la definició i, per tant, en la representació de la imatge.

Cal destacar que la conversió d'una imatge vectorial a una de mapa de bits és un procés relativament senzill de fer, mentre que el procés invers, és a dir, el de passar una imatge de mapa de bits a vectorial és molt més complex i laboriós.

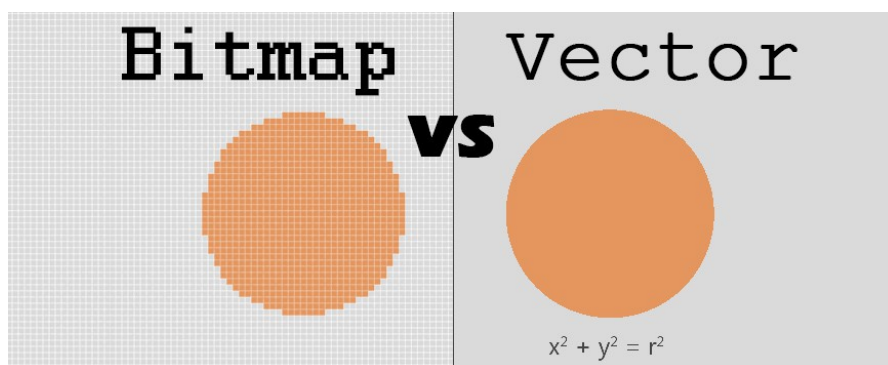


Figura 5. Fotografia de mapa de bits vs imatge vectorial

5.2. Píxel

La paraula 'píxel' ve de l'acrònim del concepte en anglès 'Picture Element', que es defineix com la part més petita i homogènia en color que compona una imatge.

En aquest apartat es veurà d'una forma més detallada què és un píxel i les seves característiques bàsiques, ja que les imatges que tractarem seran imatges digitals de mapa de bits, és a dir, imatge de píxels que és una estructura o fitxer de dades que representa una reixeta rectangular de punts de color (píxels) que es pot visualitzar en una pantalla d'ordinador.

5.3. Resolució d'una imatge

Totes les imatges digitals estan formades per una matriu rectangular de píxels.

Pel que fa a les imatges digitals de mapa de bits, la resolució de la imatge (és a dir la quantitat de detalls que s'hi poden observar en aquesta) s'expressa normalment amb dos nombres enters:

- El primer d'aquests nombres es refereix al nombre de columnes de píxels que té la imatge, o dit d'una altra forma, els píxels que té la imatge pel que fa a la seva amplada.
- El segon nombre correspon a la quantitat de files de píxels, és a dir, els píxels de la imatge pel que fa a la seva alçada.

Aquesta resolució pot ser expressada fent servir el nombre total de píxels de la imatge, que s'obté del producte dels dos nombres enters anteriors, els que fan referència a les columnes i les files de píxels.

Així doncs, per exemple, una imatge amb 1920 columnes i 1080 files de píxels, s'expressaria com a 1920x1080, o amb un únic nombre com a 2.073.600 píxels (que és el producte del 1920 per 1080), és a dir, 2 megapíxels.

5.4. Profunditat i models de color

Les característiques més importants que defineixen un píxel són la seva posició dins aquesta matriu rectangular que la forma i el seu color.

Quan es parla de profunditat de color s'entén la forma de codificació de cadascun dels píxels que conformen una imatge, gràcies a la qual es pot obtenir una completa gamma de tonalitats.

En les imatges de mapa de bits cadascun dels píxels és codificat de manera individual mitjançant un conjunt de bits d'una longitud determinada.

Ara bé, per poder convertir la informació numèrica que emmagatzema un píxel en un color, a més de la profunditat i brillantor del seu color (la mida en bits del píxel), s'ha de conèixer el model de color.

El mode més utilitzat en els monitors dels ordinadors és el RGB. Aquesta sigla fa referència a les paraules angleses *red* , *green* i *blue* (vermell, verd i blau).

Aquest model és un model cromàtic que consisteix en la representació dels diferents colors a partir d'una combinació d'aquests tres colors primaris, així doncs, depenent de la quantitat d'un i altre color el resultat obtingut variarà.

A mode d'exemple, el color violeta s'obté de la combinació del vermell i el blau, variant les quantitats utilitzades de vermell i de blau s'aconsegueix una tonalitat o altra de violeta.

Al mode RGB és freqüent que s'usin 8 bits per representar la proporció de cadascun dels tres components primaris.

Així doncs quan una de les components val 0 vol dir que no intervé a la mescla, en cas contrari quan el seu valor val $2^8 - 1$ (255) significa que hi intervé oferint el màxim de la seva tonalitat.

Tenint això en compte a continuació es mostren alguns exemples:

(255,0,0) és vermell

(0,255,0) és verd

(0,0,255) és blau

(255,255,255) és blanc

6. Tractament d'imatges

En aquest apartat s'exposarà el procés que s'ha de seguir a l'hora de fer l'estudi d'una imatge i, més en concret, el que hauria de fer el programa informàtic si s'optés per la primera proposta dels objectius particulars, realitzar la mesura de les partícules de manera automàtica a través del contorn de cadascuna d'aquestes.

El tractament d'imatges té a veure amb l'adquisició, transmissió, processament i representació de les imatges.

Les tècniques de processament d'imatges s'utilitzen per millorar l'aparença visual de les imatges per l'observador i per preparar convenientment el contingut de cara a la percepció per part de les màquines i programes.

En el cas particular d'aquest treball, estem parlant d'una aplicació a un àmbit científic, pel que les passes que s'hauran de seguir són les següents:

- 1) Capturar la imatge
- 2) Fer un procés de millora
- 3) Determinar els elements a mesurar
- 4) Executar la mesura
- 5) Emmagatzemar les mesures i fer processos gràfics i estadístics amb aquestes dades.

D'aquesta manera, per seguir les passes anteriors, és de vital importància conèixer en profunditat les diverses àrees en les que es divideix el tractament digital de imatges.

Les principals àrees del tractament digital de imatges són:

- Captura o adquisició de la imatge que s'ocupa dels diferents camins per l'obtenció d'imatges.
- Realç i millora, tècniques que s'usen per millorar l'experiència visual de les imatges, per restaurar o per recuperar les imatges degradades.

- Segmentació, que s'ocupa de dividir les imatges en regions o àrees significatives d'interès per l'estudi que es vol fer.
- Extracció de característiques. Aquesta fase s'ocupa de la detecció i localització d'entitats geomètriques simples i complexes. Això engloba entitats simples com poden ser línies i punts fins a geometries més complexes com corbes.

6.1. Realç o millora de la imatge [10]

El propòsit d'aquesta fase és millorar la qualitat de l'aparença visual de la imatge per l'observador.

És un procés subjectiu realitzat habitualment d'una forma interactiva amb l'usuari que pot personalitzar algunes de les seves característiques.

La selecció dels mètodes apropiats i l'elecció dels paràmetres adequats depenen de la qualitat de la imatge original i de l'aplicació.

Existeixen una gran quantitat de transformacions o operacions que es poden realitzar sobre les imatges per dur a terme aquest procés. Hi ha diversos criteris per classificar aquestes operacions.

Freqüentment es divideix en les següents quatre categories:

- Transformacions puntuals: el resultat d'una operació puntual depèn únicament del nivell de gris d'entrada d'un punt.

Les operacions típiques puntuals inclouen la manipulació dels píxels. La binarització o la umbralització entre d'altres, formen part d'aquest procés.

- Transformacions locals: a diferència del cas anterior, en aquest cas els valors d'entrada de diversos píxels veïns contribueixen al resultat del píxel de sortida.

Moltes operacions són locals, per exemple, el suavitzat.

- Transformacions globals: l'*input* d'entrada en aquest tipus de transformacions és el total de dades de la imatge que contribueix al resultat de sortida.

Aquestes operacions globals es realitzen sovint en el domini de la freqüència. Un exemple és

la compressió de imatges, on prenent el total d'una imatge com a entrada s'obté una imatge comprimida a la sortida.

- Transformacions geomètriques: el resultat de la transformació depèn de les diferents posicions dels nivells de gris en la imatge d'entrada.

Exemples d'aquest tipus d'operacions són la rotació i la translació, els canvis d'escala i la rectificació.

6.1.1. Soroll d'una imatge [9]

Quan s'obté una imatge digital i aquesta es veu una mica distorsionada, es diu que la imatge té soroll.

El soroll és el conjunt de senyals estranyes i no desitjades que sorgeixen en un sistema de comunicació o en qualsevol transmissió de so, degut a interferències, a defectes dels dispositius, etc.

El soroll de una imatge digital és la variació aleatòria de la brillantor o el color en les imatges digitals causat pels dispositius d'entrada i que no es correspon amb la realitat.

Això normalment és degut a que píxels aïllats de la imatge en qüestió presenten valors diferents dels valors reals.

Els tipus de soroll que es poden donar a les imatges digitals són molt variats, a continuació s'exposen els més importants:

- Additiu: $g(x,y) = f(x,y) + n(x,y)$, on $g(x,y)$ és el resultat de la distorsió de la imatge original $f(x,y)$ pel soroll $n(x,y)$. El tipus de soroll additiu més usual és el soroll gaussià.
- Impulsional: freqüentment originat per sensors. Pot ser modelat com $g(x,y) = (1-p)f(x,y) + p*i(x,y)$ on $i(x,y)$ és el soroll impulsional i p pertany a $\{0,1\}$.
- Multiplicatiu: el soroll d'aspecte granulat de les imatges d'ecografies i ràdars és bàsicament multiplicatiu. Es pot modelar amb l'expressió $g(x,y) = f(x,y)*m(x,y)$, on $m(x,y)$ és el soroll multiplicatiu.

6.1.2 Soroll gaussià

Quan es produeix soroll gaussià, el valor final del píxel és el valor real més una certa quantitat d'error, tal i com s'ha vist a l'apartat anterior als sorolls additius. Es pot descriure com una variable gaussiana que segueix el patró d'una distribució normal.

El soroll gaussià sol ser causat degut a components electrònics tals com sensors o digitalitzadors i produeix petites variacions a la imatge.



Figura 6. Imatge original sense soroll



Figura 7. Imatge amb soroll gaussià

6.1.3 Soroll de sal i pebre

En aquest cas es tracta d'un soroll del tipus impulsionar.

El valor que pren el píxel no té relació amb el valor real sinó que assumeix valors molt alts o molt baixos.

Així doncs, pot prendre valors màxims (sal) o mínims (pebre).



Figura 8. Imatge original sense soroll



Figura 9. Imatge original amb soroll de sal i pebre

6.1.4. Filtrat de la imatge

Les tècniques de filtrat, són mètodes usats per ressaltar o suprimir, de forma selectiva, informació continguda en una imatge a diferents escales espacials.

Això es fa tant per destacar elements que siguin d'interès de la pròpia imatge com per ocultar valors anòmals.

El procés de filtrar consisteix en l'aplicació a cadascun dels píxels de la imatge d'una matriu de mida $N \times N$ (normalment és una matriu 3×3 però pot ésser més gran) formada per nombres enters i que genera un nou valor mitjançant una funció del valor original i dels píxels del voltant.

Els filtres més utilitzats són els de passa baix (els quals s'encarreguen de suavitzar la imatge), de passa alt (que augmenten el contrast), filtres direccionals (tenen la funció de detectar estructures en la imatge que segueixen una determinada direcció) i els de detecció de contorns (permeten identificar i aïllar objectes amb propietats homogènies dins la imatge).

6.1.4.1. Filtres de passa baix

L'objectiu que persegueixen aquests filtres és suavitzar la imatge.

Aquest tipus de filtres resulten especialment útils quan se suposa que la imatge presenta una gran quantitat de soroll i es vol eliminar.

Existeixen diferents tipus que són:

- Filtre de la mitjana, assignen al central la mitjana de tots els píxels inclosos a la finestra.
- Filtre de la mediana, té l'avantatge de que el valor final del píxel es un valor real present a la imatge i no un promig.

Gràcies a això, es redueix l'efecte borros que tenen les imatges que han estat tractades amb un filtre de mitjana.

L'inconvenient que presenta és que el seu càlcul és més complex ja que s'han d'ordenar els diferents valors que apareixen als píxels inclosos a la finestra i determinar el seu valor central.

- Filtres adaptatius.
- Filtres gaussians, anomenats d'aquesta manera perquè simulen una distribució gaussiana.

El valor màxim apareix al píxel central i disminueix cap als extrems tant ràpid com ho indiqui el paràmetre de desviació típica s .

Quant menor sigui aquest paràmetre de desviació, major serà la velocitat la que ho farà.

El resultat serà un conjunt de valors entre 0 i 1.

6.1.4.2. Filtres de passa alt

Aquest tipus de filtre pretén ressaltar les zones de major variabilitat eliminant el que seria la component mitja, precisament la que detectem els filtres de passa baix.

D'altra banda, part de la resposta de cada píxel es troba "contaminada" per la dels píxels veïns.

Existeixen diversos mètodes:

- Sostracció de la mitjana: si es considera que un filtre de passa baix serveix per ressaltar components a gran escala eliminant la variabilitat local, si a la imatge original se li resta el resultat del passa baix s'aconsegueix ressaltar aquesta variabilitat local.
- Filtres basats en les derivades: la segona derivada en el cas d'una imatge serveix per informar com són els canvis, més o menys pronunciats, que es produeixen entre píxels contigus.

Dins d'aquest tipus de filtres es troba el filtre laplacià recomanat pel realç de trets lineals en entorns urbans.

Una altra opció per ressaltar els elements de major variabilitat és restar a la imatge original l'obtinguda per mitjà del filtre laplacià.

6.1.4.3. Filtres direccionals

S'utilitzen per a detectar perfils que segueixen una determinada direcció en l'espai, ressaltant el contrast entre els píxels situats a ambdós costats del perfil.

6.1.4.4 Filtres de detecció de contorns

Un dels més utilitzats i conegut és la detecció de contorns mitjançant l'algoritme de Canny.

Es veurà de manera detallada a 6.1.5.

6.1.5. Detecció de contorns, l'algoritme de Canny [2]

En l'àmbit del processament d'imatges i, de manera més concreta, en la mesura de les partícules que es volen mesurar, la detecció dels contorns d'una imatge és de vital importància pel reconeixement d'objectes.

L'algoritme de Canny és usat per detectar contorns d'una imatge i és considerat com un dels millors algoritmes per dur a terme aquest procés.

Aquest algoritme està basat en l'ús de màscares de convolució i la primera derivada, els punts del contorn són com zones de píxels en les que existeix un canvi brusc del nivell de gris.

Així doncs, en aquest mètode la primera derivada pren com a valor 0 en totes les regions on no varia la intensitat i un valor constant en tota la transició de intensitat.

Per tant, un canvi de intensitat es manifesta com un canvi molt pronunciat en la primera derivada, característica que és usada precisament per detectar un contorn i en la que es basa l'algoritme de Canny.

L'algoritme de Canny es divideix en tres parts ben diferenciades que es mostren a continuació, també es pot considerar una quarta part en aquest procés que seria el tancament de contorns oberts per mitjà de l'algoritme de Deriche y Cocquerez.

6.1.5.1. Obtenció del gradient

En aquesta fase es calcula la magnitud i orientació del vector gradient en cada píxel.

Per a l'obtenció del gradient el primer que es realitza es l'aplicació d'un filtre gaussià a la imatge original amb l'objectiu de suavitzar-la i tractar d'eliminar tot el possible soroll que pugui existir a la imatge.

Ara bé, s'ha de vigilar a l'hora de fer aquest suavitzat, ja que si es realitza un suavitzat excessiu es perdran dades de la imatge provocant que el resultat final que s'obtingui no sigui bo, pel que ha de triar-se el nivell del suavitzat més apropiat per la imatge que estiguem tractant.

Un cop suavitzada adequadament la imatge, per cada píxel s'obtenen la magnitud i mòdul del gradient, obtenint així dues imatges.

6.1.5.2. Supressió no màxima

En aquest apartat es pretén reduir l'amplada dels contorns, obtinguts a partir del gradient, fins a aconseguir contorns d'un píxel d'amplada.

Les dues imatges obtingudes a la fase anterior serveixen d'entrada per generar una imatge amb els contorns prims.

El procediment que es segueix consisteix en considerar quatre direccions corresponents a les orientacions de 0° , 45° , 90° i 135° respecte l'eix horitzontal.

Per a cada píxel es troba la direcció que millor s'aproxima a la direcció de l'angle del gradient.

6.1.5.3. Histèresis de llindar a la supressió no màxima

En aquest tercer pas s'aplica una funció d'histèresis basada en dos llindars, amb aquest mecanisme el que es pretén és reduir la possibilitat d'aparició de contorns falsos o contorns no desitjats.

La imatge obtinguda a l'apartat anterior sol contenir màxims locals creats pel soroll. Una solució per eliminar aquest soroll és la histèresis del llindar.

El procés consisteix en prendre la imatge obtinguda del pas anterior, prendre la orientació dels punts del contorn de la imatge i prendre dos llindars, el primer més petit que el segon.

Per a cada punt de la imatge s'ha de localitzar el següent punt del contorn no explorar que sigui major que el segon llindar.

A partir d'aquest punt seguir les cadenes de màxims locals connectats entre ambdós direccions perpendiculars a la normal del contorn sempre que siguin majors al primer llindar.

Així es marquen tots els punts explorats i s'emmagatzema la llista de tots els punts en el contorn connectat.

Gràcies a aquest procés s'aconsegueixen eliminar les unions en forma de Y dels segments que conflueixen en un sol punt.

6.1.5.4. Exemple gràfic de l'aplicació de l'algoritme Canny

En aquest apartat s'il·lustra de forma visual el resultat d'aplicar l'algoritme de Canny a una fotografia.



Figura 10. Imatge original



Figura 11. Imatge després d'aplicar l'algoritme de Canny

6.2. Aplicació dels algorismes anteriors a la mesura de les nanopartícules [6-8]

Tot l'estudi fet anteriorment sobre el tractament d'imatges es vol aplicar a la mesura de les nanopartícules de les imatges preses per microscòpia electrònica.

Per fer-ho s'ha realitzat un codi en Python.

També s'ha fet ús de llibreries com *OpenCV*, llibreria que ja du incorporat els filtres i algorismes que es volen aplicar a la imatge (concretament un filtre gaussià per eliminar el soroll que pugui haver a la imatge i, posteriorment, l'algoritme de Canny per detectar els contorns de cadascuna de les partícules per poder-les mesurar més endavant una per una per separat).

El resultat però, no ha estat del tot satisfactori, motiu pel qual s'ha destriat aquesta opció coma solució al problema.

La complexitat que presenten les imatges que es volen tractar dificulta aquest procés de

detecció de contorns i fa que el resultat final no sigui l'esperat.

A continuació es mostra un exemple simple en els que els contorns es troben perfectament delimitats, i després, el resultat obtingut a l'hora d'aplicar l'algoritme al programa a través de diferents valors d'histèresis.

Tal i com es pot veure, quan es tracta la imatge de les partícules, apareixen molts contorns "falsos" i que a més, si són tractats per separat, es pot observar com molts cops comprenen més d'una partícula.



Figura 12. Imatge original de cercles

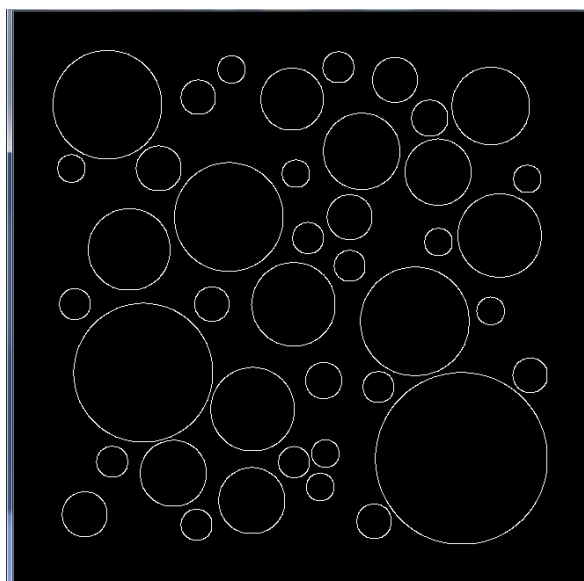


Figura 13. Imatge de cercles després d'aplicar l'algoritme de Canny a través del programa dissenyat

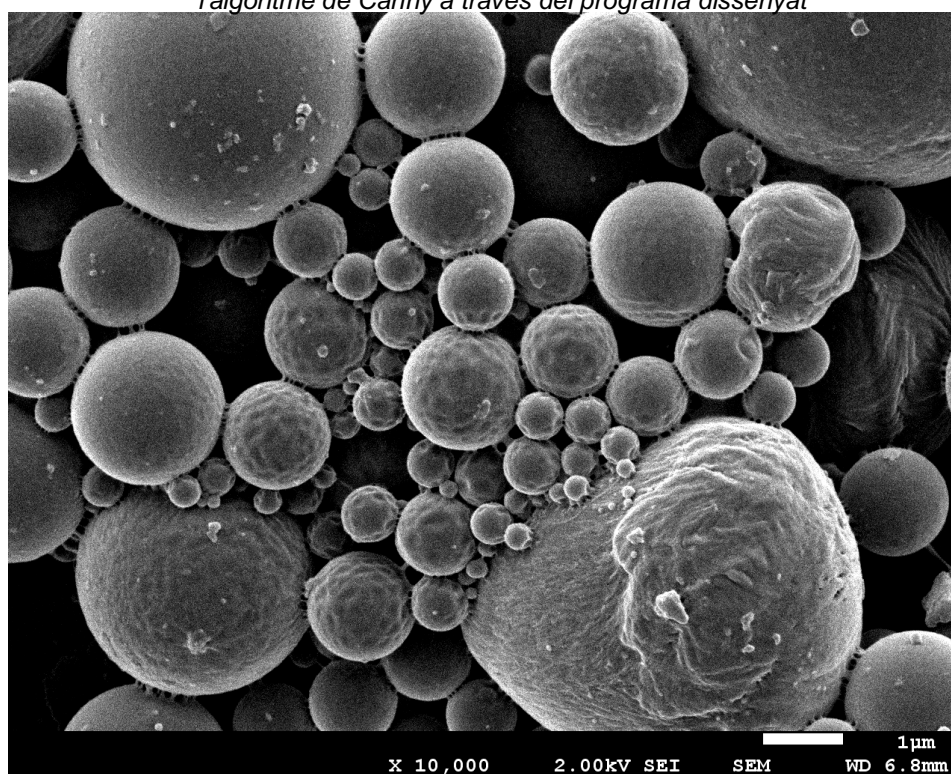


Figura 14. Imatge original de partícules

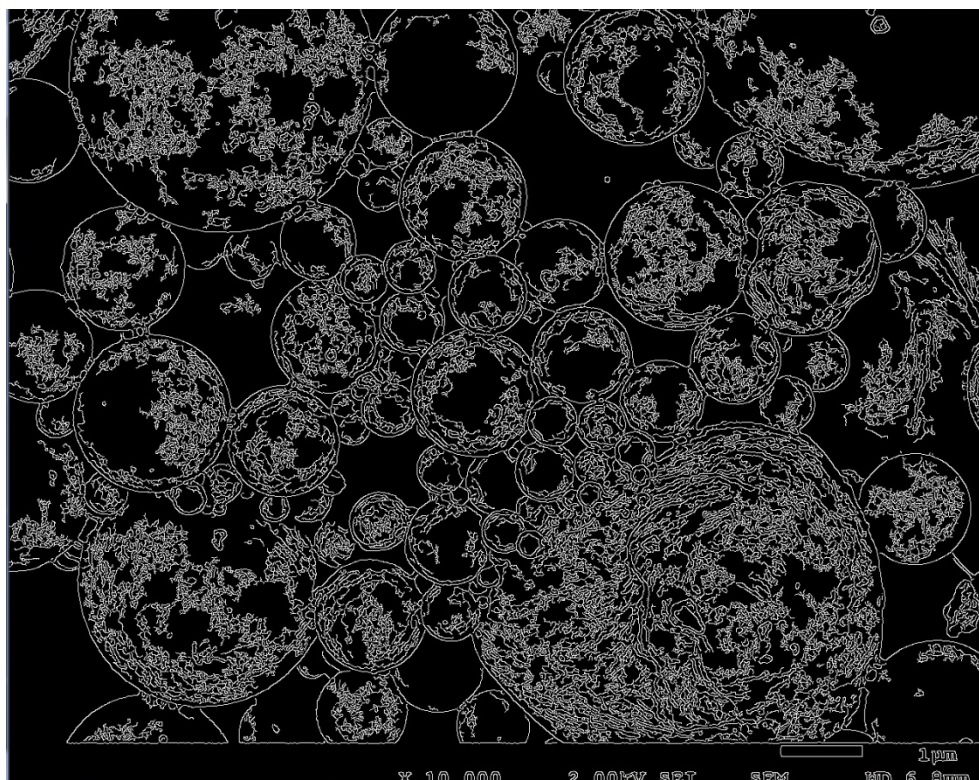


Figura 15. Imatge de partícules després d'aplicar l'algoritme de Canny a través del programa dissenyat amb una histèresis de (10, 150)

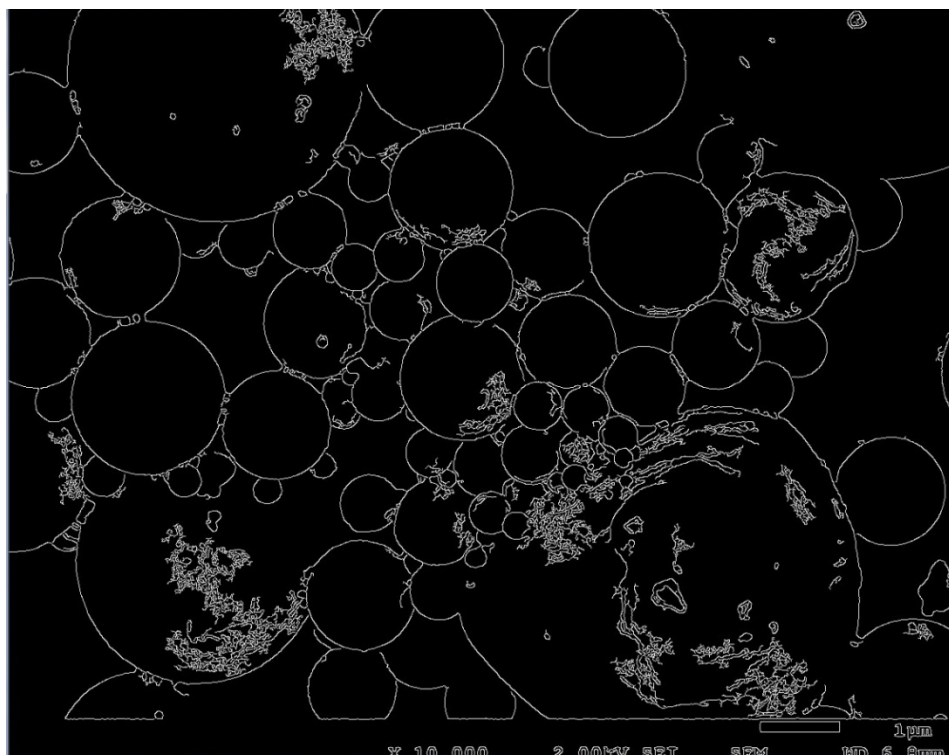


Figura 16. Imatge de partícules després d'aplicar l'algoritme de Canny a través del programa dissenyat amb una histèresis de (10, 350)

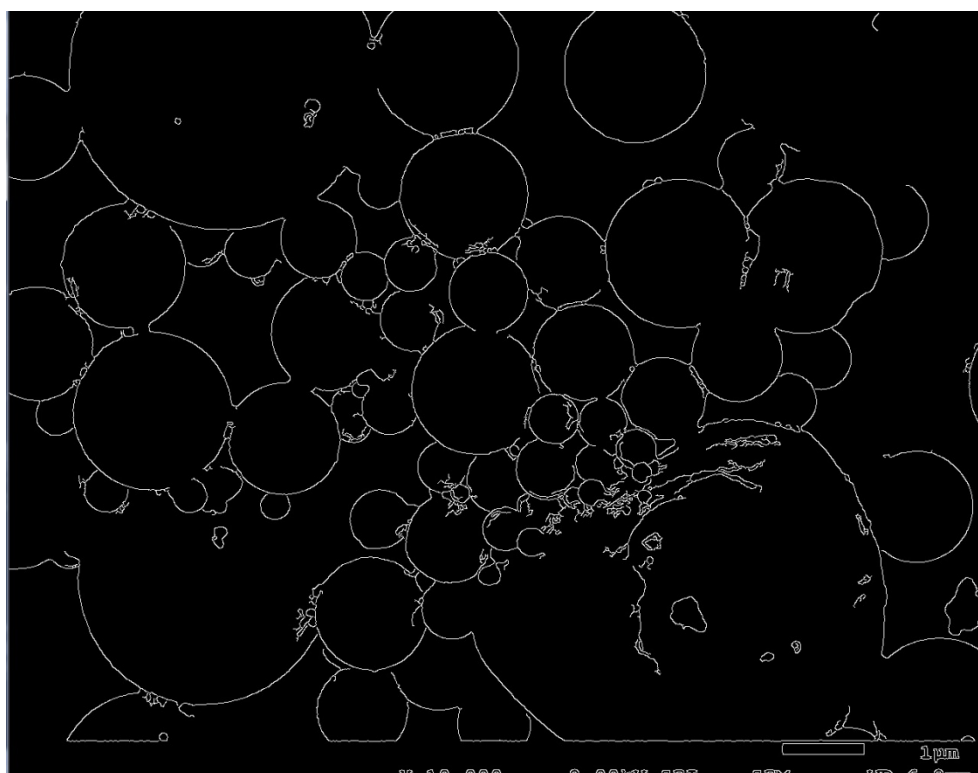


Figura 17. Imatge de partícules després d'aplicar l'algoritme de Canny a través del programa dissenyat amb una histèresis de (50, 450)

7. Traçat de cercles

Com ja s'ha vist als apartats anteriors la solució que es pretenia buscar mitjançant el tractament de la imatge amb filtres i algorismes per delimitar el seu contorn no ha estat del tot satisfactori.

Així doncs, s'ha procedit a seguir la segona via plantejada als objectius del projecte que tracta de determinar la mida de les partícules a partir de la indicació de tres punts de cadascuna de les partícules per part de l'usuari del programa.

S'ha arribat a aquesta solució ja que les partícules d'àcid polilàctic (PLA) que es volen mesurar es poden considerar com a esfèriques.

Tenint en compte que, una esfera sobre un pla (en aquest cas la imatge captada per microscòpia electrònica) descriu un cercle, simplifica molt la problemàtica.

Ara bé, el paràmetres principal que defineix una circumferència (contorn d'un cercle) és el radi (o diàmetre), o directament relacionat amb aquest, la longitud del perímetre que conforma la circumferència ($L = 2\pi \cdot \text{radi}$).

El problema resulta quan no es coneixen ni un paràmetre ni l'altre (de fet són les incògnites que es volen trobar).

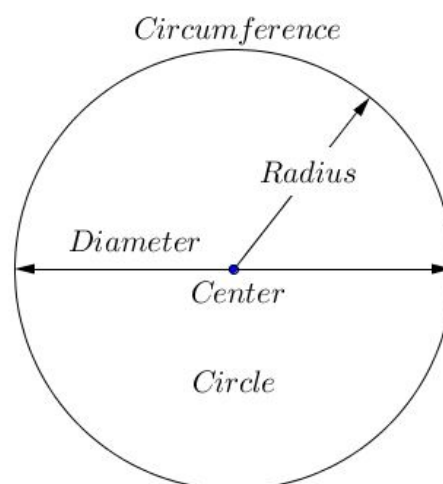


Figura 18. Circumferència

Per la complexitat de la imatge el procés de mesura purament automàtic no ha resultat satisfactori.

Segons sembla fins ara els programes de detecció i definició d'imatges no han resolt algunes deficiències en la interpretació del pixelat de les mateixes, així per exemple programes de protecció contra la pornografia arriben a bloquejar imatges de sorra a les quals confonen amb nus de cossos humans.

A partir d'aquesta conclusió es descarta, de moment, l'automatització total del procés i es conclou que serà necessària la intervenció de l'ull humà per a la interpretació correcta de los contorns de les imatges.

L'ull de l'usuari sí que és capaç de reconèixer el contorn de cadascuna d'aquestes partícules esfèriques.

Aquí farem entrar en joc una propietat important de les circumferències: tres punts donats en un plànol descriuen una i només una circumferència possible.

Així doncs, si l'usuari indica tres punts del contorn d'una de les partícules, el programa informàtic per mitjà de càlculs a través de fórmules implementades al seu codi intern, pot generar la circumferència que forma el seu contorn i, per tant, determinar la mida del seu diàmetre, i a partir d'aquesta dada es poden realitzar els càlculs pertinents sobre la resta de dimensions de la nanopartícula.

A la imatge que es presenta a continuació es pot veure de manera gràfica com es calcula el centre de la circumferència en qüestió.

Un cop la coordenada del centre està correctament determinada, el radi és únicament la distància del centre a un dels tres punts prèviament dibuixats (distància que, òbviament, és la mateixa a qualsevol dels tres punts).

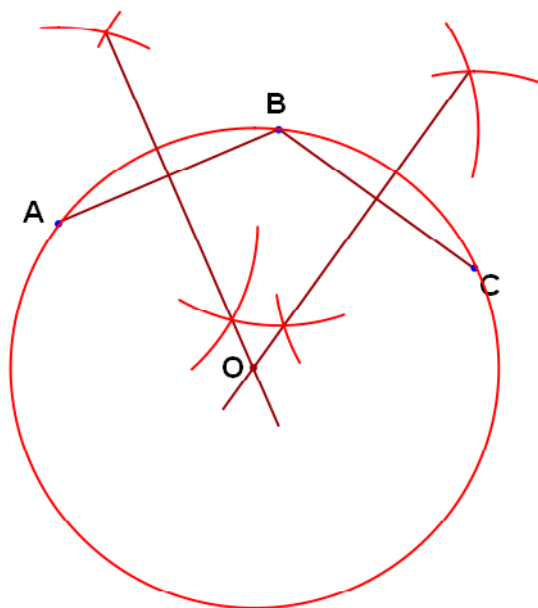


Figura 19. Traçat d'una circumferència de manera gràfica a partir de tres punts

Ara bé, el programa informàtic no calcula aquestes circumferències de manera gràfica com s'ha vist anteriorment.

Té en el seu codi implementades les següents fórmules matemàtiques que determinen de manera precisa la coordenada (x,y) del centre C de la circumferència.

A les fórmules que es mostren a continuació les coordenades (x_1, y_1) , (x_2, y_2) i (x_3, y_3) corresponen respectivament a les coordenades dels punts 1, 2 i 3 (indicats per l'usuari del programa) de la circumferència que es vol traçar. (C_x, C_y) fan referència a la coordenada del centre C de la circumferència.

$$m_a = \frac{y_2 - y_1}{x_2 - x_1}$$

$$m_b = \frac{y_3 - y_2}{x_3 - x_2}$$

$$C_x = \frac{m_a \cdot m_b \cdot (y_1 - y_3) + m_b \cdot (x_1 + x_2) - m_a \cdot (x_2 + x_3)}{2 \cdot (m_b - m_a)}$$

$$C_y = -\frac{1}{m_a} \cdot \left(C_x - \frac{x_1 + x_2}{2}\right) + \frac{y_1 + y_2}{2}$$

Un cop determinada la coordenada del centre a través d'aquestes fórmules matemàtiques, el següent pas és que el programa dibuixi aquesta circumferència.

8. Resultats

8.1. Codi del programa [3-5]

A continuació es mostra el codi del programa que consta aproximadament d'unes 140 línies.

Inicialment es mostra la importació de les llibreries que s'usen al programa. Després s'inicialitzen les variables que formen el programa.

Les funcions que es troben (per ordre) corresponen a les funcionalitats de: crear i guardar l'arxiu de text amb les dades mesurades, el canvi de mode del programa a mode "dibujar escala", el canvi de mode del programa a mode "dibujar contorno partícules", dibuixar l'element desitjat (segons quin sigui el mode treball), redimensionar la imatge per ajustar-la a la finestra que s'obre i obrir la imatge desitjada.

Les darreres línies de codi que apareixen corresponen a la creació de la finestra del programa i dels menús que el formen.

```
from tkinter import *  
  
from tkinter import font  
  
from tkinter import PhotoImage  
  
from tkinter import Canvas  
  
from tkinter import NW  
  
from tkinter import Menu  
  
from tkinter import filedialog  
  
from tkinter import messagebox  
  
from PIL import Image, ImageTk  
  
from math import sqrt
```



```
mode = 'inicio'

lista_puntos_escala = []

lista_puntos_particula=[]

pixels_barra = 1

unidad_escala = 'µm'

medida_escala = 1

n=0

lista_diametro_particulas = []

particulas = ''

centro_x = None

centro_y = None


def guardar_archivo_texto():

    f = filedialog.asksaveasfile(mode='w', defaultextension=".txt")

    if f is None:

        return

    f.write(particulas)

    f.close()


def dibujar_escala():

    global mode, medida_escala, unidad_escala

    mode = 'escala'

    messagebox.showinfo('Mensaje', 'Modo de trabajo: dibujar escala')
```

```
def dibujar_circulos():

    global mode

    mode = 'circulos'

    messagebox.showinfo('Mensaje', 'Modo de trabajo: dibujar contorno
partículas')

def dibujar(event):

    global mode, lista_puntos_escalas, lista_puntos_particula,
pixels_barra, n, lista_diametro_particulas, centro_x, centro_y,
particulas, medida_escalas, unidad_escalas

    pt = (event.x, event.y)

    if mode == 'escalas':

        canvas.create_oval(event.x - 1, event.y - 1, event.x + 1,
event.y + 1, fill = 'blue')

        lista_puntos_escalas.append(pt)

        if len(lista_puntos_escalas) == 2:

            #print('Dibujar escalas')

            canvas.create_line(lista_puntos_escalas[0][0],
lista_puntos_escalas[0][1], event.x, lista_puntos_escalas[0][1])

            pixels_barra = abs(lista_puntos_escalas[0][0] - event.x)

            lista_puntos_escalas = []

    if mode == 'circulos':
```

```

        canvas.create_oval(event.x - 2, event.y - 2, event.x + 2,
event.y + 2, fill = 'red')

    lista_puntos_particula.append(pt)

    if len(lista_puntos_particula) == 3:

        n=n+1

        x1=lista_puntos_particula[0][0]

        y1=lista_puntos_particula[0][1]

        x2=lista_puntos_particula[1][0]

        y2=lista_puntos_particula[1][1]

        x3=lista_puntos_particula[2][0]

        y3=lista_puntos_particula[2][1]

        #Ponemos esta condición para asegurarnos que ma y mb tienen
un valor definido

        if x2-x1 != 0 and x3-x2 != 0:

            ma=(y2-y1)/(x2-x1)

            mb=(y3-y2)/(x3-x2)

            centro_x          =          (ma*mb*(y1-y3)+mb*(x1+x2)-
ma*(x2+x3))/(2*(mb-ma))

            centro_y = (-1/ma)*(centro_x-(x1+x2)/2)+(y1+y2)/2

            centro = (int(centro_x), int(centro_y))

            rad = sqrt(pow(x1 - centro_x, 2) + pow(y1 - centro_y,
2))

            canvas.create_oval(centro[0] - int(rad),centro[1] -
int(rad), centro[0]+ int(rad), centro[1] + int(rad), width = 4)

            lista_puntos_particula = []

            lista_diametro_particulas.append(2*rad)

```

```

        Helvfont = font.Font(family="Helvetica", size=7,
weight="bold")

        etiqueta_particula = Label(ventana, text= str(n),
font=Helvfont).place(x = centro[0] - 1 ,y = centro[1] - 1, height=6,
width=6)

particulas=particulas + 'Particula' + str(n) + ' ' +
str(2*rad*medida_escal/(pixels_barra)) + ' ' + unidad_escal + '\n'

if x2-x1 == 0 or x3-x2 == 0:

    lista_puntos_particula = []

    n=n-1

def redimensionar(imagen_original):

    o_size = imagen_original.size #Tamaño original de la imagen

    f_size = (800, 800)

    factor = min(float(f_size[1])/o_size[1],
float(f_size[0])/o_size[0])

    width = int(o_size[0] * factor)

    height = int(o_size[1] * factor)

    return (imagen_original.resize((width, height), Image.ANTIALIAS))

def abrir():

    global imagenL

    ventana.filename=filedialog.askopenfilename()

    ruta=ventana.filename

```

```

imagen = Image.open(ruta)

imagenL = ImageTk.PhotoImage(redimensionar(imagen))

canvas.create_image(0,0,anchor=NW,image=imagenL)


ventana =Tk ()

ventana.geometry("800x800+0+0")

ventana.title("Sistema de medición de partículas")

#Creas tu canvas y lo posicionas en tu ventana

canvas = Canvas (ventana, height=800, width=800)

canvas.pack()

# Variable global que va a contener la imagen

imagenL = None

barraMenu=Menu(ventana)

#crear los
menús .....

mnuArchivo=Menu(barraMenu)

mnuEscala=Menu(barraMenu)

mnuDibujar=Menu(barraMenu)

mnuDatos=Menu(barraMenu)

#crear los comandos de los
menús.....

mnuArchivo.add_command(label="Abrir",command = abrir)

mnuArchivo.add_separator()

mnuArchivo.add_command(label="Salir",command=ventana.destroy)

```

```
mnuEscala.add_command(label = "Dibujar escala", command =
dibujar_escalas)

mnuDibujar.add_command(label = "Contorno partícula a partir de 3
puntos", command= dibujar_circulos)

mnuDatos.add_command(label = 'Generar fichero de datos', command =
guardar_archivo_texto)

barraMenu.add_cascade(label="Archivo",menu=mnuArchivo)

barraMenu.add_cascade(label="Escala",menu=mnuEscala)

barraMenu.add_cascade(label="Dibujar",menu=mnuDibujar)

barraMenu.add_cascade(label="Datos",menu=mnuDatos)

ventana.config(menu=barraMenu)

canvas.bind("<Button-1>", dibujar)

ventana.mainloop()
```

A continuació s'adjunten les captures de pantalla del codi del programa per poder visualitzar-lo millor i poder-ne distingir correctament les paraules reservades pròpies de Python.

```

1  from tkinter import *
2  from tkinter import font
3  from tkinter import PhotoImage
4  from tkinter import Canvas
5  from tkinter import NW
6  from tkinter import Menu
7  from tkinter import filedialog
8  from tkinter import messagebox
9  from PIL import Image, ImageTk
10 from math import sqrt
11
12
13 mode = 'inicio'
14 lista_puntos_escalas = []
15 lista_puntos_particula = []
16 pixels_barra = 1
17 unidad_escalas = 'µm'
18 medida_escalas = 1
19 n = 0
20 lista_diametro_particulas = []
21 particulas = ''
22 centro_x = None
23 centro_y = None
24
25 def guardar_archivo_texto():
26     f = filedialog.asksaveasfile(mode='w', defaultextension=".txt")
27     if f is None:
28         return
29     f.write(particulas)
30     f.close()

```

Figura 20. Codi del programa (1)

```

31
32
33 def dibujar_escalas():
34     global mode, medida_escalas, unidad_escalas
35     mode = 'escalas'
36     messagebox.showinfo('Mensaje', 'Modo de trabajo: dibujar escalas')
37
38
39 def dibujar_circulos():
40     global mode
41     mode = 'circulos'
42     messagebox.showinfo('Mensaje', 'Modo de trabajo: dibujar contorno partículas')
43
44
45 def dibujar(event):
46     global mode, lista_puntos_escalas, lista_puntos_particula, pixels_barra, n, lista_diametro_particulas, centro_x, centro_y, particulas, medida_escalas
47     pt = (event.x, event.y)
48     if mode == 'escalas':
49         canvas.create_oval(event.x - 1, event.y - 1, event.x + 1, event.y + 1, fill = 'blue')
50         lista_puntos_escalas.append(pt)
51         if len(lista_puntos_escalas) == 2:
52             #print('Dibujar escalas')
53             canvas.create_line(lista_puntos_escalas[0][0], lista_puntos_escalas[0][1], event.x, lista_puntos_escalas[1][1])
54             pixels_barra = abs(lista_puntos_escalas[0][0] - event.x)
55             lista_puntos_escalas = []
56
57
58     if mode == 'circulos':
59         canvas.create_oval(event.x - 2, event.y - 2, event.x + 2, event.y + 2, fill = 'red')

```

Figura 21. Codi del programa (2)



Figura 22. Codi del programa (3)

```

60     lista_puntos_particula.append(pt)
61     if len(lista_puntos_particula) == 3:
62         n=n+1
63         x1=lista_puntos_particula[0][0]
64         y1=lista_puntos_particula[0][1]
65         x2=lista_puntos_particula[1][0]
66         y2=lista_puntos_particula[1][1]
67         x3=lista_puntos_particula[2][0]
68         y3=lista_puntos_particula[2][1]
69         #Ponemos esta condición para asegurarnos que ma y mb tienen un valor definido
70         if x2-x1 != 0 and x3-x2 != 0:
71             ma=(y2-y1)/(x2-x1)
72             mb=(y3-y2)/(x3-x2)
73             centro_x = (ma*mb*(y1-y3)+mb*(x1+x2)-ma*(x2+x3))/(2*(mb-ma))
74             centro_y = (-1/ma)*(centro_x-(x1+x2)/2)+(y1+y2)/2
75             centro = (int(centro_x), int(centro_y))
76             rad = sqrt(pow(x1 - centro_x, 2) + pow(y1 - centro_y, 2))
77             canvas.create_oval(centro[0] - int(rad),centro[1] - int(rad), centro[0]+ int(rad), centro[1] + int(rad), width = 4)
78             lista_puntos_particula = []
79             lista_diametro_particulas.append(2*rad)
80             Helvfont = font.Font(family="Helvetica", size=7, weight="bold")
81             etiqueta_particula = Label(ventana, text= str(n), font=Helvfont).place(x = centro[0] - 1 ,y = centro[1] - 1, height=6, width=6)
82             #print ('Particula' + str(n) + ' ' + str(2*rad*medida_escal/(pixels_barra))+ unidad_escal)
83             particulas=particulas + 'Particula' + str(n) + ' ' + str(2*rad*medida_escal/(pixels_barra)) + ' ' + unidad_escal + '\n'
84             #print ('Dibujar circulos')
85         if x2-x1 == 0 or x3-x2 == 0:
86             lista_puntos_particula = []
87             n=n-1

```

Figura 23. Codi del programa (4)

```

89
90 def redimensionar(imagen_original):
91     o_size = imagen_original.size #Tamaño original de la imagen
92     f_size = (800, 800)
93     factor = min(float(f_size[1])/o_size[1], float(f_size[0])/o_size[0])
94     width = int(o_size[0] * factor)
95     height = int(o_size[1] * factor)
96     return (imagen_original.resize((width, height), Image.ANTIALIAS))
97
98 def abrir():
99     global imagenL
100     ventana.filename=filedialog.askopenfilename()
101     ruta=ventana.filename
102     imagen = Image.open(ruta)
103     imagenL = ImageTk.PhotoImage(redimensionar(imagen))
104     canvas.create_image(0,0,anchor=NW,image=imagenL)
105
106 ventana =Tk ()
107 ventana.geometry("800x800+0+0")
108 ventana.title("Sistema de medición de partículas")

```

Figura 24. Codi del programa (5)

```
109 #Creas tu canvas y lo posicionas en tu ventana
110 canvas = Canvas (ventana, height=800, width=800)
111 canvas.pack()
112
113 # Variable global que va a contener la imagen
114 imagenL = None
115 barraMenu=Menu(ventana)
116
117 #crear los menús .....
118 mnuArchivo=Menu(barraMenu)
119 mnuEscala=Menu(barraMenu)
120 mnuDibujar=Menu(barraMenu)
121 mnuDatos=Menu(barraMenu)
122
123 #crear los comandos de los menús.....
124 mnuArchivo.add_command(label="Abrir",command = abrir)
125 mnuArchivo.add_separator()
126 mnuArchivo.add_command(label="Salir",command=ventana.destroy)
127
128 mnuEscala.add_command(label = "Dibujar escala", command = dibujar_escala)
129
130 mnuDibujar.add_command(label = "Contorno partícula a partir de 3 puntos", command= dibujar_circulos)
131
132
133 mnuDatos.add_command(label = 'Generar fichero de datos', command = guardar_archivo_texto)
134
135 barraMenu.add_cascade(label="Archivo",menu=mnuArchivo)
136 barraMenu.add_cascade(label="Escala",menu=mnuEscala)
137 barraMenu.add_cascade(label="Dibujar",menu=mnuDibujar)
```

Figura 25. Codi del programa (6)

```
138 barraMenu.add_cascade(label="Datos",menu=mnuDatos)
139 ventana.config(menu=barraMenu)
140
141 canvas.bind("<Button-1>", dibujar)
142
143 ventana.mainloop()
144
```

Figura 26. Codi del programa (7)

8.2. Esquema general del programa

A continuació es mostra l'esquema general del programa i el camí que segueix l'usuari des que obre el programa i la imatge les partícules de la qual desitja mesurar, fins que ha acabat de mesurar les partícules i ha exportat les dades d'aquestes a un programa extern, en aquest cas Excel. Un cop es tenen les dades a Excel es poden manipular i realitzar els càlculs matemàtics i estadístics pertinents per tal d'extreure conclusions fiables i que siguin d'utilitat.

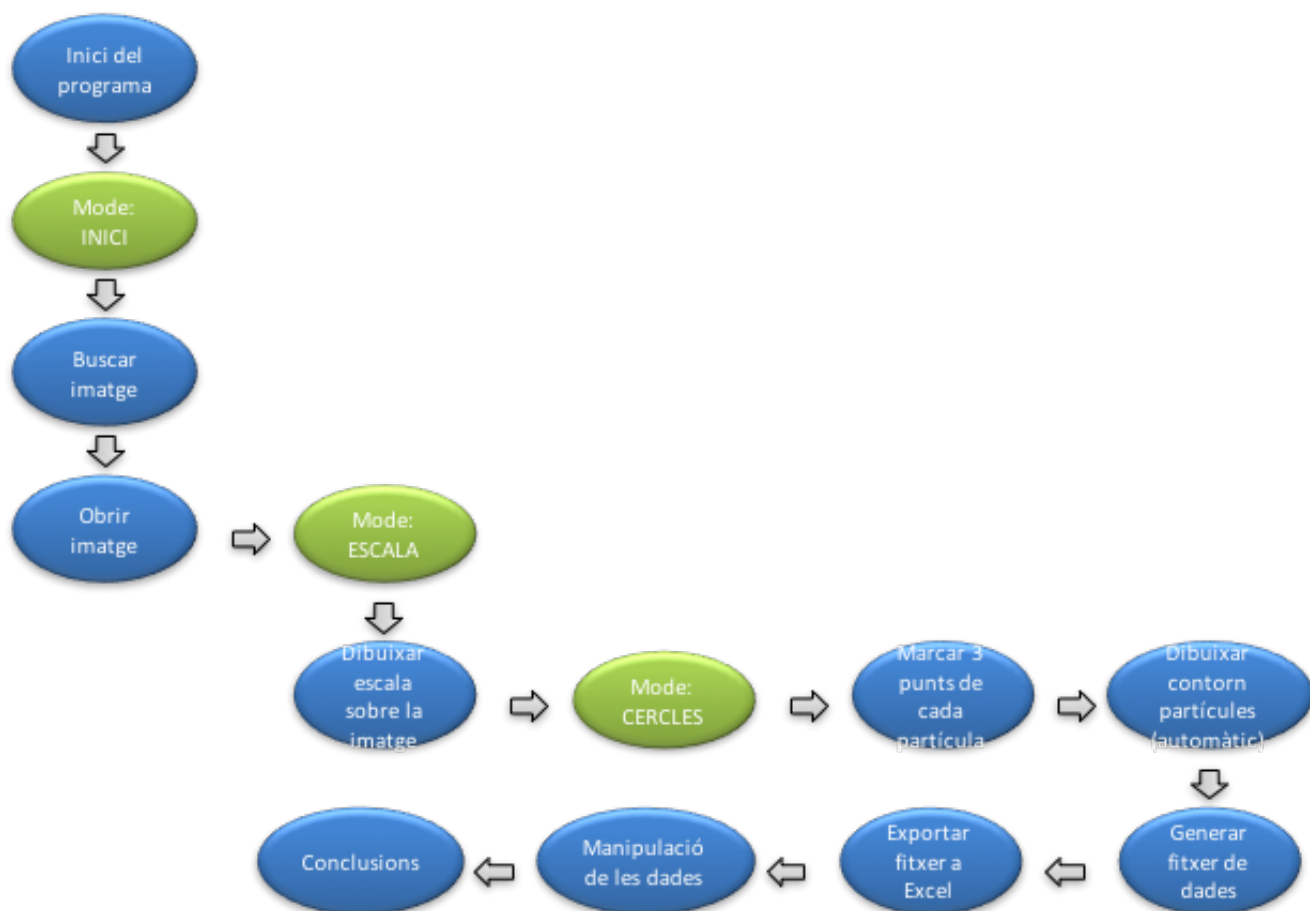


Figura 27. Esquema general del programa informàtic dissenyat

8.3. Algoritme del programa

En aquest apartat es pot observar l'algoritme bàsic que fa servir el programa l'hora de dibuixar les diferents figures geomètriques.

Aquestes són punts, línies per determinar l'escala i cercles per determinar els diàmetres de les partícules de la imatge.

```
INICIALITZACIÓ del programa
```

```
Mode del programa en mode: 'inicio'
```

```
SELECCIONAR Archivo - Abrir
```

```
    Seleccionar la imatge que es vol mesurar amb la finestra que s'obre
```

```
SELECCIONAR Escala - Dibujar Escala
```

```
    Mode del programa canvia a mode: 'escala'
```

```
    Marcar punt de l'extrem de la barra de l'escala (primer un extrem i després  
    l'oposat):
```

```
        Afegir punt a la lista_puntos_escala
```

```
        Si nombre de punts en lista_puntos_escala == 2
```

```
            Escala dibuixada i determinada pel programa (nombre de píxels  
            de la barra de l'escala)
```

```
        Fi
```

```
    Fi
```

```
SELECCIONAR Dibujar - Contorno partícula a partir de 3 puntos
```

```
    Mode del programa canvia a mode: 'circuitos'
```

```
    Marcar amb el ratolí punt del contorn de la partícula:
```

```
        Afegir punt a la lista_puntos_particula
```

```
        Si nombre de punts en lista_puntos_particula == 3:
```

```
            Contorn de la partícula dibuixada a partir dels 3 punts
```

```
            Afegir diàmetre la partícula mesurada a  
            lista_diametro_particulas
```

```
lista_puntos_particula = [ ] (llista buida)
```

```
Fi
```

```
Fi
```

SELECCIONAR Datos - Generar fichero de datos

Es genera el fitxer de dades amb el string emmatgzemat a
lista_diametro_particulas

Guardar el fitxer de text (extensió .txt) amb el nom desitjat i al directori
desitjat a través de la finestra que s'obre

```
Fi
```

8.4. Funcionalitats del programa i guia per l'usuari

En aquest apartat es poden veure les funcionalitats que inclou el programa.

8.4.1. Obrir imatge

Aquesta funcionalitat, com el seu nom indica, permet obrir la imatge que l'usuari desitgi estudiar i analitzar a través de la barra d'eines situada a la part superior de la finestra.

Per accedir a aquesta opció, cal anar a l'opció 'Archivo' i a la pestanya que s'obre, fer clic amb el ratolí sobre l'opció 'Abrir'.

Un cop s'ha accedit a aquesta opció, el programa mostrarà una nova finestra en la que apareixerà, per defecte, el directori en el que s'està treballant. A través d'aquesta finestra, l'usuari pot escollir la imatge que vulgui (ja sigui d'aquest directori de treball o un altre).

Després de triar la imatge, el programa la mostrarà a la finestra principal, substituint el fons blanc que apareix inicialment en el moment en el que s'executa el programa.

En quant al format de les imatges que pot obrir, el programa funciona amb .jpg, .jpeg, .gif, .png. Altres tipus de formats d'imatge no s'han testejat.

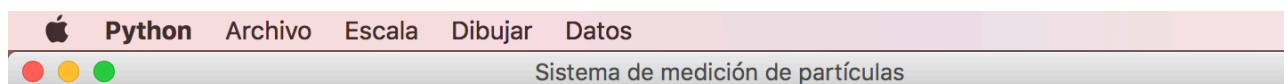


Figura 28. Pantalla inicial del programa

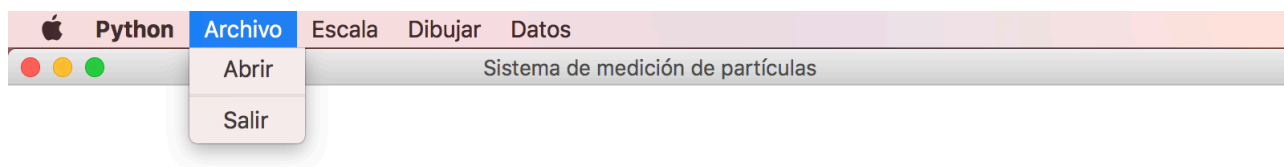


Figura 29. Menú "Archivo" del programa

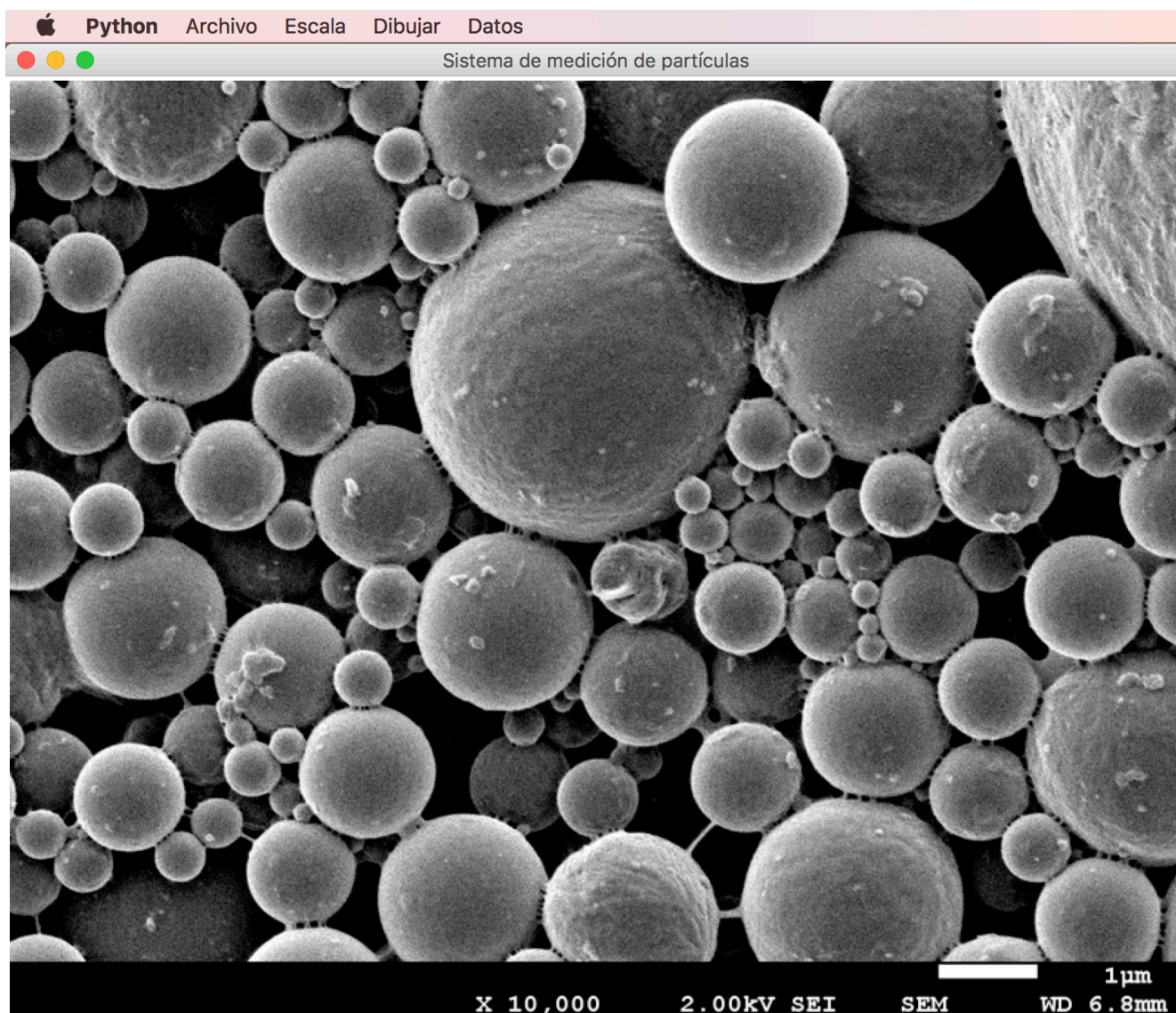


Figura 30. Pantalla del programa un cop oberta la imatge desitjada

8.4.2. Dibujar escala

Gràcies a aquesta funcionalitat, el programa és capaç de determinar a escala real el diàmetre de les partícules que es volen mesurar.

Aquesta funcionalitat és imprescindible, ja que el programa per tractar la imatge treballa en píxels, i a l'usuari l'interessa la mesura de la partícula en unitat mètrica (mm, μm , nm...).

Per fer això, l'usuari haurà d'accedir a l'opció 'Escala - Dibujar escala' situada a la barra d'eines de la part superior de la finestra principal.



Figura 31. Menú "Escala" del programa

A continuació, el programa avisarà a través d'un missatge emergent (una nova finestra s'obrirà), de que el mode de treball ha canviat al mode 'dibujar escala'.

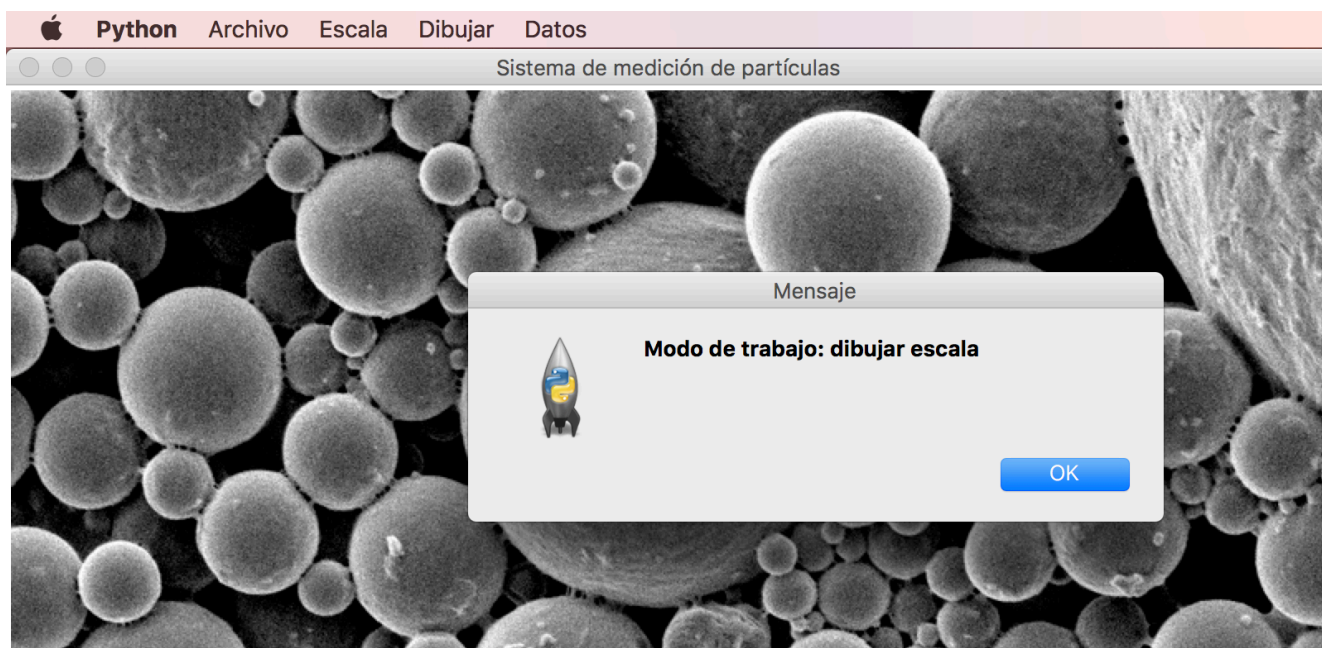


Figura 32. Missatge emergent un cop activat el mode "dibujar escala"

Amb aquest mode activat, l'objectiu és dibuixar una línia recta a través de dos punts escollits per l'usuari. Aquests dos punts hauran de correspondre a l'extrem esquerre i l'extrem dret de la

barra de l'escala que apareix a la part inferior de les imatges. Per dibuixar un punt, s'han de fer un clic amb el botó esquerre del ratolí.

Quan ja s'han dibuixat els dos punts, el programa automàticament dibuixa el segment que els uneix.

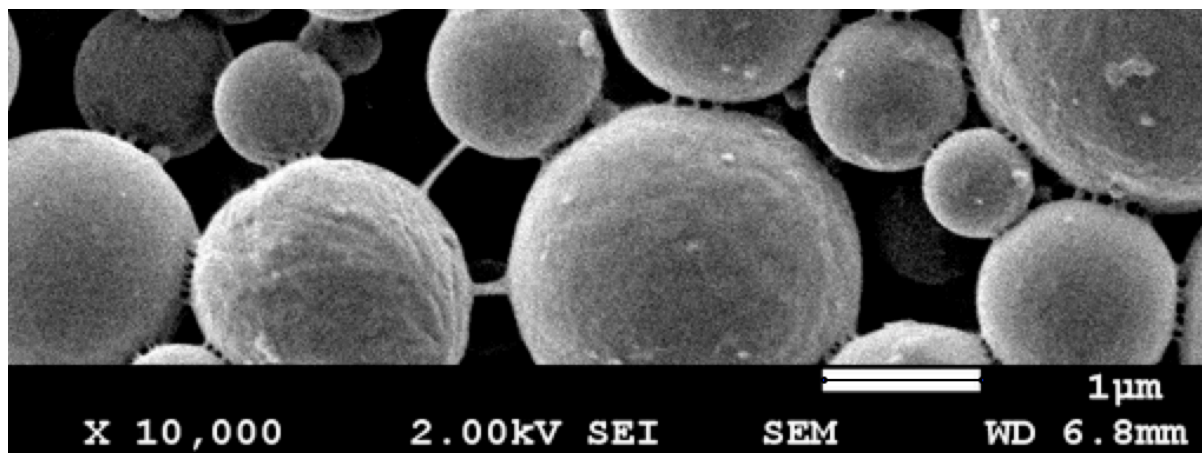


Figura 33. Segment de l'escala dibuixat a partir de dos punt de la barra

Per traçar la línia només considera la coordenada x, de manera que la coordenada y és només la del primer punt dibuixat, i per tant, la línia que apareix sempre és horitzontal.

A partir d'aquesta línia dibuixada, el programa calcula el nombre de píxels que la forma i troba l'equivalència entre 1 unitat de mesura i aquest nombre de píxels.

Així doncs, per exemple, si l'escala està formada per 50 píxels, l'equivalència amb unitats de mesura reals és $1/50 = 0,02$ (si l'escala marca 1 μm). Això vol dir, que cada píxel equival a 0,02 μm.

Com és lògic, tot això s'haurà d'executar un cop l'usuari ha obert la imatge que vol estudiar.

Quan ja s'ha determinat de manera correcta l'escala, és el moment de dibuixar el contorn de les partícules i mesurar-les.

8.4.3. Dibuixar contorn i determinar diàmetre de les partícules

Amb aquesta funcionalitat, es poden crear circumferències a partir de la determinació de tres punts de l'usuari.

A l'apartat 7. , s'han mostrat les fórmules matemàtiques que el programa té implementades al seu codi per tal de poder traçar les circumferències.

Per poder fer-ho, cal que l'usuari vagi altre cop a la barra d'eines situada a la part superior de la finestra principal i es dirigeixi a les opcions 'Dibujar - Contorno partícula a partir de 3 puntos'.



Figura 34. Menú "Dibujar" del programa

Quan es selecciona aquesta opció, apareix un altre missatge emergent en una nova finestra indicant que el mode ha canviat al mode 'dibujar contorno partículas'.

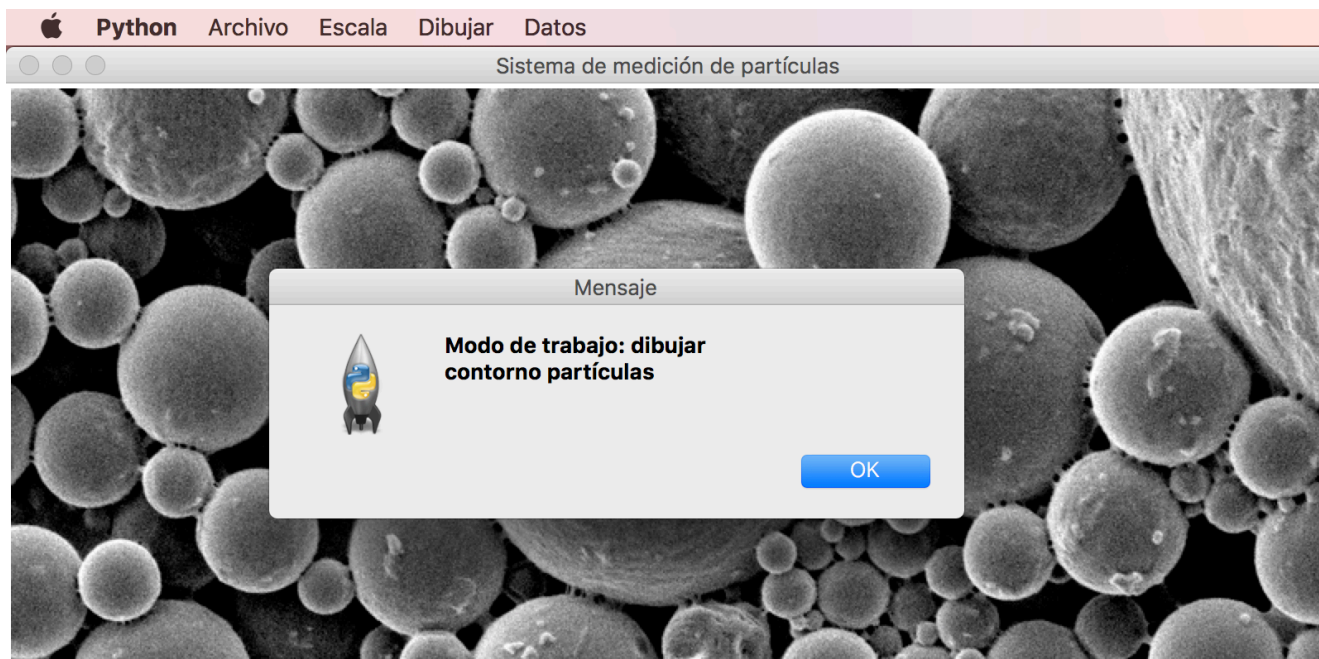


Figura 35. Missatge emergent un cop activat el mode "dibujar contorno partículas"

Ara, l'usuari està en disposició de traçar el contorn de les partícules. Per fer-ho, haurà d'indicar al programa tres punts del seu contorn. Com en el cas anterior, cadascun d'aquests punts es marcaran fent un clic amb el botó esquerre del ratolí.

Un cop s'hagin dibuixat els tres punts, el programa de manera automàtica dibuixa la circumferència, i, a través de l'escala, calcula el seu diàmetre en unitats de mesura i no en píxels.

A la següent imatge es pot veure el traçat del contorn de 6 partícules, cadascuna numerada al seu centre.

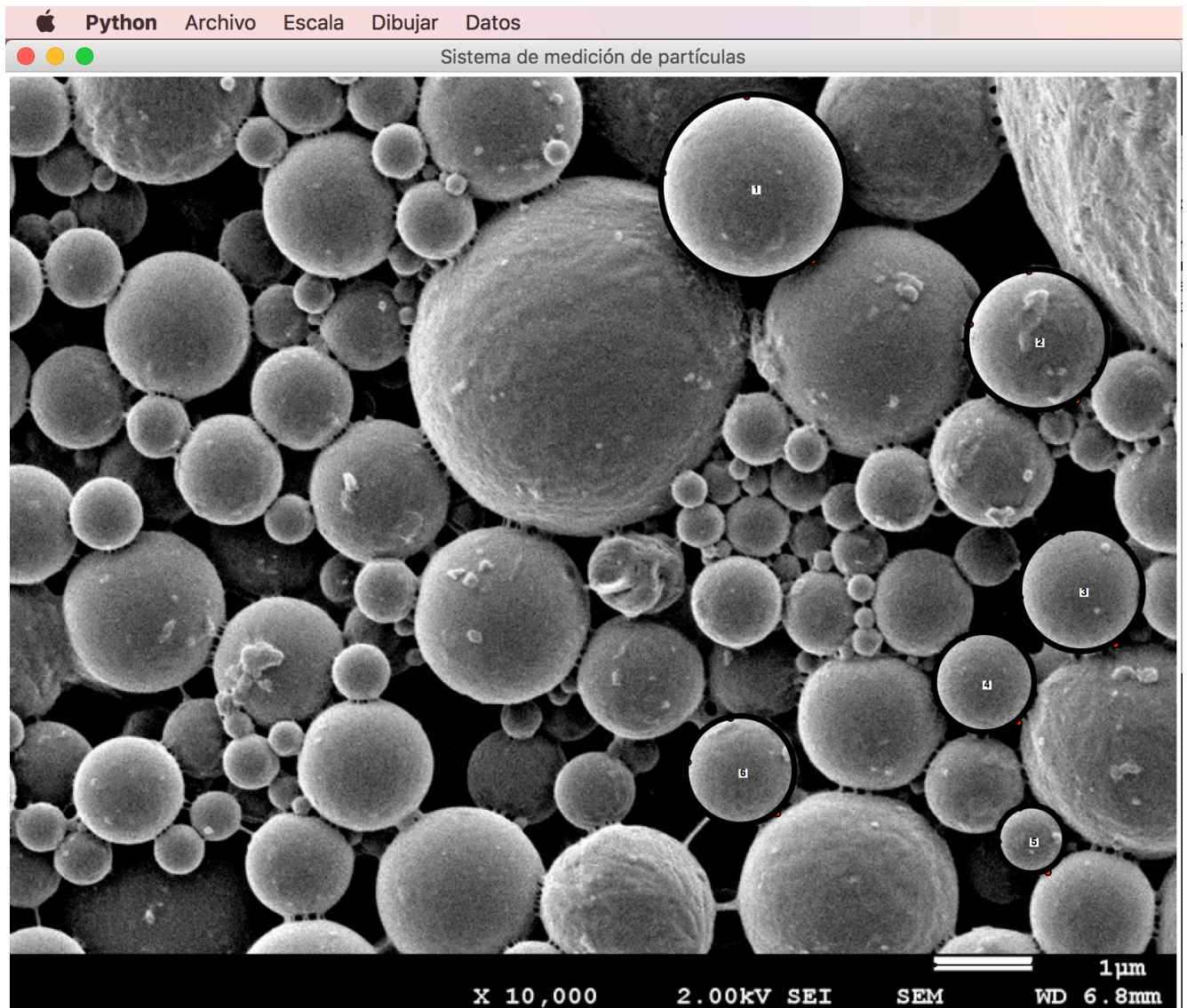
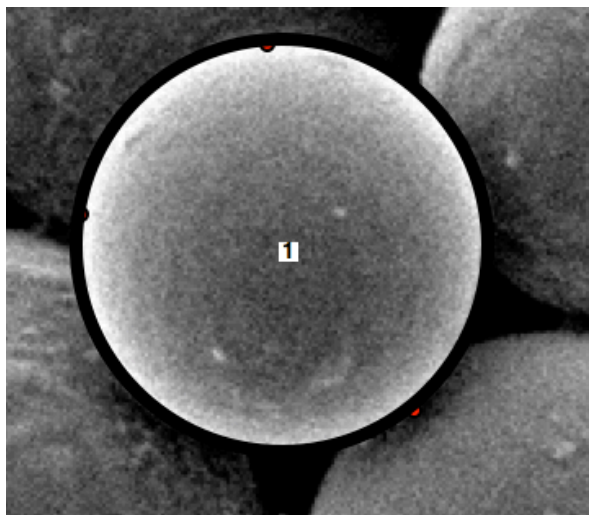
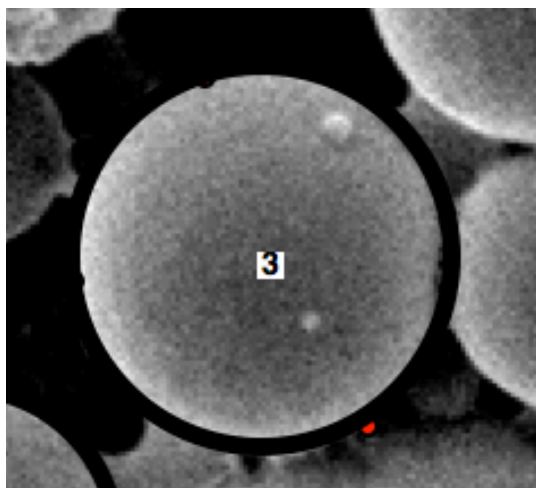
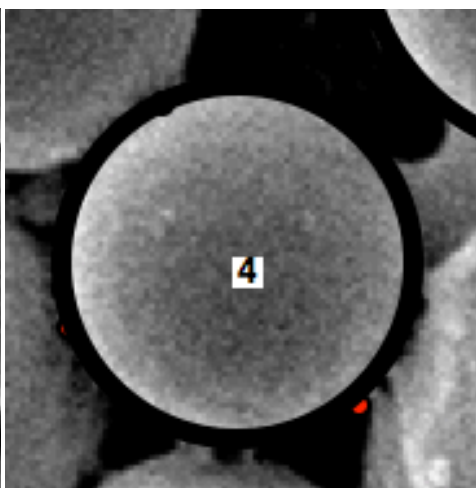


Figura 36. Contorn de sis partícules delimitat i mesurat pel programa

8.4.4. Numeració de les partícules

Al mateix temps que el programa dibuixa el contorn de la partícula, el programa numera aquesta partícula amb un número al seu centre (de manera que la primera partícula que es mesuri apareixerà un '1', la segona un '2' i així successivament).

*Figura 37. Primera partícula mesurada**Figura 38. Segona partícula mesurada**Figura 39. Tercera partícula mesurada**Figura 40. Quarta partícula mesurada*

És una eina que s'ha implementat amb dues intencions.

La primera intenció és la més evident, i és que l'usuari sigui capaç d'identificar cadascuna de les partícules un cop es generi el fitxer de dades on apareixerà el número de partícula i el seu corresponent diàmetre.

La segona intenció està relacionada amb si l'usuari s'equivoca a l'hora de determinar algun dels punts de la partícula. Si això succeeix, la circumferència traçada serà incorrecta i per tant, la seva mesura també. Ara bé, aquesta circumferència mal traçada (perquè no s'ajusta de manera correcta a la partícula que es vol mesurar) ha estat identificada amb un número, que

posteriorment apareixerà al fitxer de dades que es generarà i que per tant, la podrà eliminar l'usuari manualment.

8.4.5. Generació del fitxer de dades

En aquest instant l'usuari ja ha obert la imatge, ha determinat l'escala de manera correcta per tal de que el programa faci amb precisió la conversió entre píxels i unitats de mesura i, per últim, ha dibuixat el contorn de les partícules que són d'interès i que apareixen a la imatge. Queda un últim pas, obtenir aquestes dades.

Per fer-ho, s'ha implementat una funcionalitat que genera de manera automàtica un fitxer de text de format .txt amb les dades.

A cadascuna de les línies d'aquest fitxer apareixerà un text (string) del tipus 'PartículaN Diàmetre_partícula' on 'N' correspon al número de partícula (és a dir, si és la partícula 1, 2, 3...) i *Diàmetre_partícula* és el diàmetre de la partícula calculat prèviament pel programa.

Per generar aquest fitxer, cal accedir a l'opció 'Datos - Generar fichero de datos'.



Figura 41. Menú "Datos" del programa

Seguidament, s'obrirà una nova finestra on es mostrarà el navegador per a que l'usuari indiqui al directori on es vol guardar aquest fitxer .txt amb les dades. També, tindrà l'opció de guardar el fitxer esmentat amb el nom que desitgi.

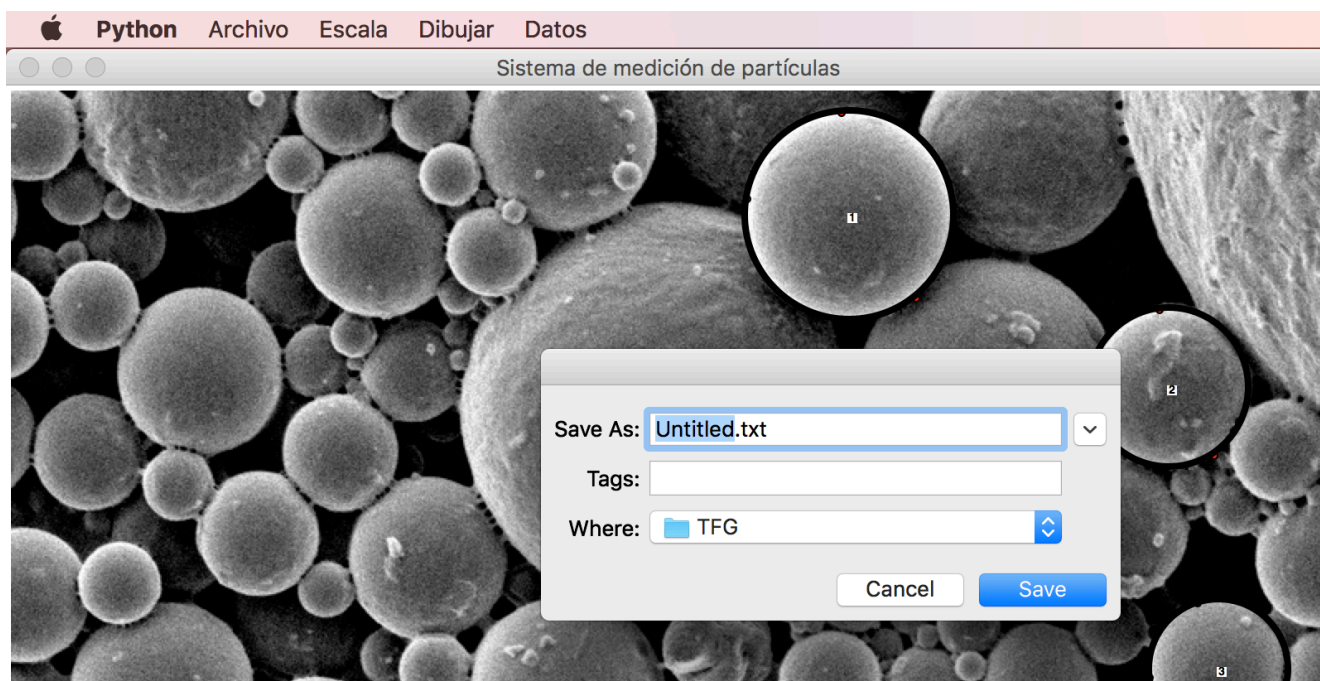


Figura 42. Finestra emergent un cop es selecciona l'opció "Generar fichero de datos"

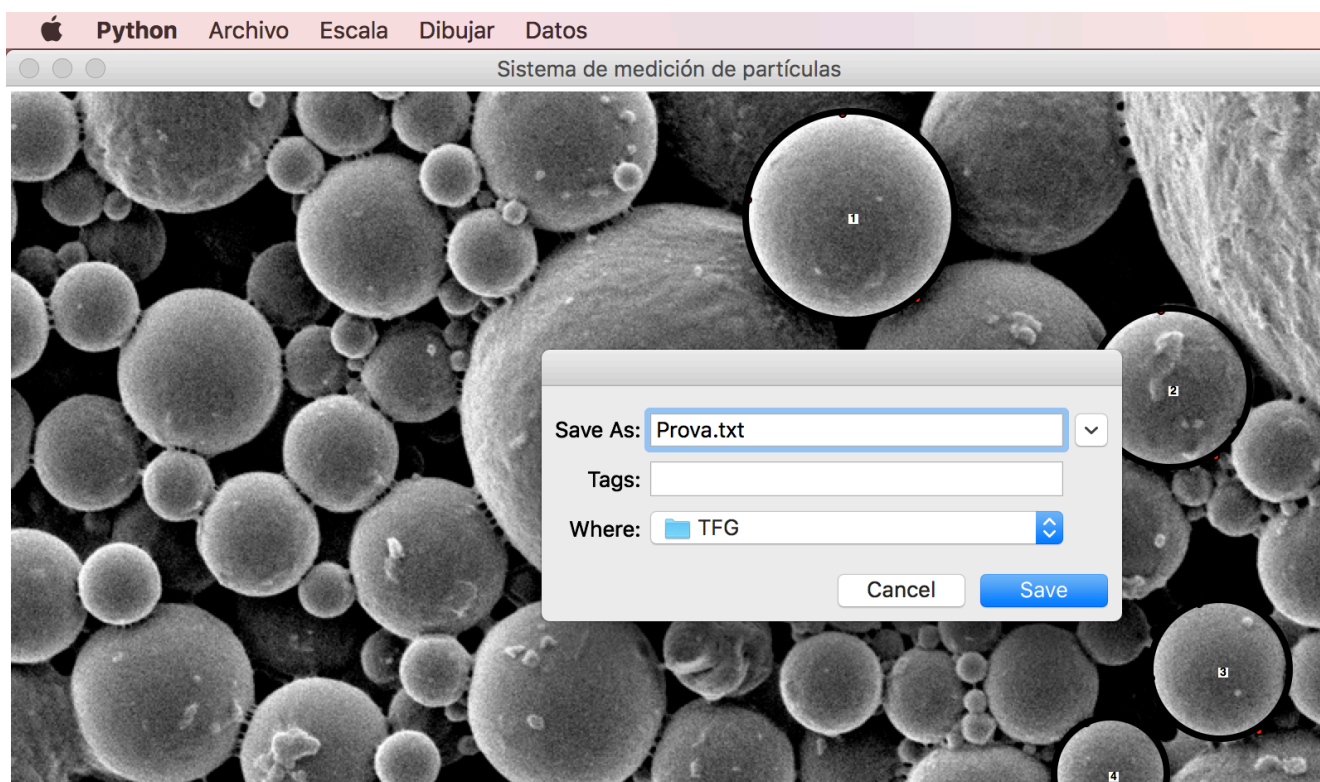


Figura 43. Nom de l'arxiu modificat correctament



Figura 44. Arxiu de text (.txt) generat a partir de les partícules mesurades

8.4.6. Manipulació de les dades

Un cop generat el fitxer .txt amb les dades, aquestes es poden copiar i enganxar a algun programa tradicional de tractament de dades com ara Microsoft Excel.

Si copiem i enganxem el contingut del fitxer a Excel, aquest apareix tot en una mateixa columna. Això inicialment suposa un problema, ja que per manipular les dades de manera correcte, interessa que el número de partícula figuri en una columna i el seu diàmetre en una altra.

Per solucionar-ho, a continuació es mostren un seguit d'imatges on s'explica de manera detallada el procediment per dur-ho a terme.

Les indicacions i imatges que es mostraran a continuació, corresponen a la versió de Microsoft Excel 2010, pot ser que el procés variï si es tracta d'una versió diferent.

Un detall important és que per determinar l'escala només s'ha introduït (de manera gràfica) el nombre de píxels de la barra. Però la seva equivalència amb unitats de mesura real, no.

Per tant, si l'escala correspon a 1, no s'ha de fer res amb les dades, però si és diferent, els valors dels diàmetres obtinguts s'hauran de multiplicar per aquesta constant (així, per exemple, si l'escala marca 5 µm, tots els diàmetres s'hauran de multiplicar per 5).

Al treballar amb Excel, això no suposa cap problema ja que multiplicar tota una columna per una constant és senzill, però és un aspecte molt important a tenir en compte ja que de no fer-ho, totes les mesures serien incorrectes.

8.5. Possibles funcionalitats i millores del programa

En aquest apartat es proposen algunes funcionalitats a considerar per implementar en un futur per tal de millorar l'ús del programa i l'experiència de l'usuari amb aquest.

8.5.1. Funció desfer

És una de les funcionalitats que facilitaria la feina per corregir els possibles errors i equivocacions de l'usuari durant el seu ús.

Com a alternativa a aquesta opció, el fet de que les partícules estiguin numerades com s'ha vist anteriorment pot facilitar la feina de l'usuari per destriar les partícules que no han estat ben mesurades (és a dir, les circumferències mal traçades).

Si no és possible implementar aquesta funció com a tal, també s'ha pensat en altres opcions com demanar a l'usuari la confirmació de considerar la circumferència traçada com a correcta (i per tant, emmagatzemar les seves dades) o de descartar-la.

8.5.2. Determinar escala a través del programa

És una altra de les funcionalitats que es té la intenció d'implementar, o, si més no, de poder millorar respecte a la versió actual del programa.

La millora passa per introduir, a través d'algun menú addicional l'escala numèrica o, en el cas de que no fos possible, a través de la mateixa consola per on s'executa el programa.

9. Estudi econòmic del projecte

Per realitzar l'estudi econòmic i el cost total, s'ha tingut en compte el nombre d'hores dedicat per dur a terme el treball, i d'altra banda el cost del material que s'ha fet servir durant la seva realització.

En quant al temps dedicat, els resultats són els següents.

Tasques realitzades	Hores dedicades
Recerca d'informació i investigació sobre tractament d'imatges	70
Recerca de documentació sobre llibreries i eines útils per realitzar el programa informàtic	80
Disseny ,realització i testeig del programa informàtic	100
Redacció de la memòria del projecte	50
TOTAL	300

Cost personal	Hores totals dedicades	Total
25€/hora	300	7.500 €

Pel que fa a l'ús del material i la seva amortització, s'ha fet ús d'un ordinador portàtil, el preu del qual és de 1150€.

Tenint en compte que la vida útil de l'ordinador correspon a 5 anys, i la quantitat d'hores d'ús a l'any és de 1800 hores, el cost que això suposa és de (s'han fet 300 hores d'ús durant la realització del treball)

$$\frac{300h \cdot 1150 \text{ €}}{5 \text{ anys} \cdot 1800 \frac{h}{\text{any}}} = 38,33 \text{ €}$$

A més, s'ha de tenir en compte el consum energètic de l'ordinador, que és de 60 W. Tenint en compte que s'ha usat 300 hores i el preu és 0,12€/kW·h

$$300h \cdot 0,060kW \cdot 0,12 \frac{\text{€}}{kW} = 2,16 \text{ €}$$

Cost personal	7.500 €
Cost del material	38,33 €
Cost energètic	2,16 €
Subtotal	7.540,49 €
IVA 21%	1.583,50 €
Total	9.123,99 €

10. Impacte ambiental del projecte

Pel que fa a l'impacte ambiental de la realització del projecte, s'han considerat la despeses energètiques provocades per l'ordinador que s'ha usat. Aquestes despeses energètiques porten associades unes emissions de CO₂ a l'atmosfera.

El factor d'equivalència que relaciona aquestes dues variables és de 1kWh = 0,308 kg de CO₂ d'emissió.

Material de treball	Potència (kW)	Temps de funcionament (h)	Consum total (kWh)
Ordinador portàtil	0,060	300	18

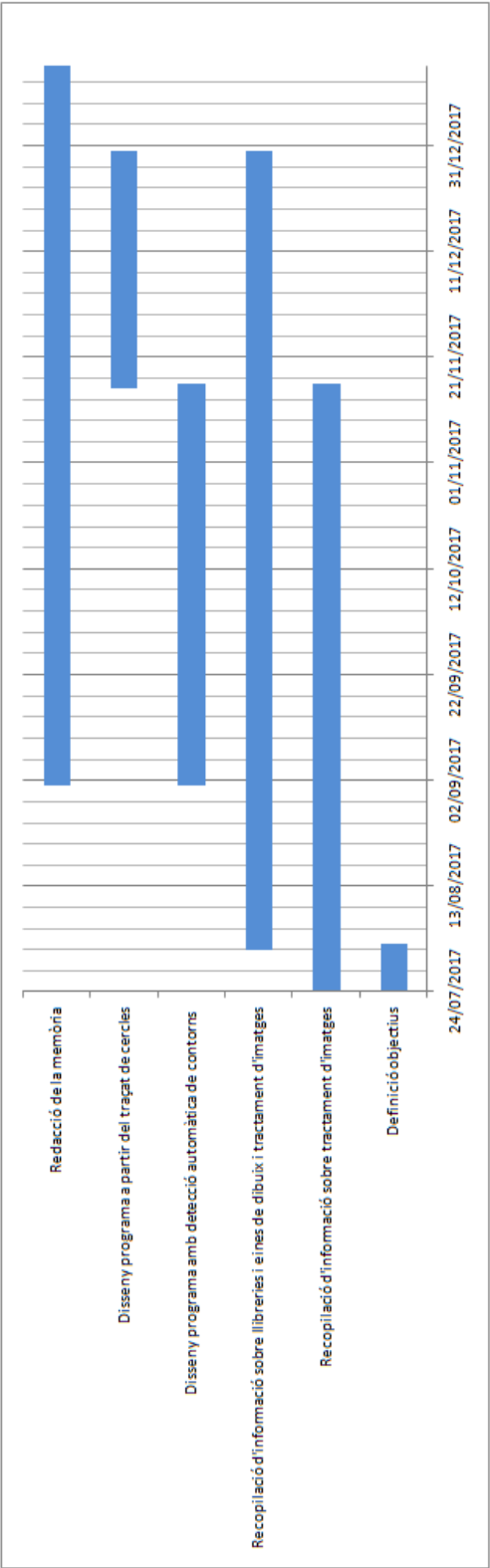
Així doncs, fent ús de la relació d'equivalència i el consum total, es pot determinar que l'emissió total de CO₂ a l'atmosfera durant la realització del treball és de 5,544 kg de CO₂.

11. Planificació

A continuació es mostra la planificació del projecte i com s'ha dividit el calendari d'execució de cadascuna de les tasques que s'han dut a terme per la realització del treball.

Tota aquesta planificació queda plasmada a la següent taula i el seu corresponent diagrama de Gantt.

Ordre	Activitat	Data d'inici	Duració (dies)	Data de finalització
1	Definició d'objectius	24/7/17	9	1/8/17
2	Recopilació d'informació sobre tractament d'imatges	24/7/17	115	15/11/17
3	Recopilació d'informació sobre llibreries i eines de dibuix i tractament d'imatges	1/8/17	151	29/12/17
4	Disseny programa amb detecció automàtica de contorns	1/9/17	76	15/11/17
5	Disseny programa a partir del traçat de cercles	15/11/17	45	29/12/17
6	Redacció de la memòria	1/9/17	137	15/1/18



Conclusions

Després de la realització del treball i investigar sobre el tractament d'imatges, es pot concloure que el camp de la "visió artificial" per la identificació de figures en el seu context és un àmbit en el que hi ha marge de millora.

L'automatització completa de segons quins processos no és possible avui dia i la solució requereix d'una certa participació de la interpretació de l'ull humà per a que el sistema pugui donar resultats de sortida siguin fiables.

En el cas concret del projecte, s'ha trobat una solució que millora el sistema actual de mesura tant en precisió com en temps.

El codi del programa consta aproximadament d'unes 140 línies. El programa, dissenyat amb Python, fa ús principalment de la llibreria tkinter per crear la interfície gràfica i fer el tractament de la imatge, entre d'altres llibreries. Les funcions que es troben (per ordre) corresponen a les funcionalitats de: crear i guardar l'arxiu de text amb les dades mesurades, el canvi de mode del programa a mode "dibujar escala", el canvi de mode del programa a mode "dibujar contorno partículas", dibuixar l'element desitjat, redimensionar la imatge per ajustar-la a la finestra que s'obre i obrir la imatge desitjada. Finalment, es procedeix a la creació de la finestra del programa i dels menús que el formen.

Amb el programa dissenyat únicament amb 3 clics de ratolí es pot dibuixar amb una bona precisió el contorn circular d'una partícula i determinar el seu diàmetre, enfront a l'escassa precisió i excessiva lentitud de l'actual procés manual de mesura.

D'aquesta manera, tot i que el procés no és del totalment automàtic, el programa facilita aquest procés de mesura amb creus, malgrat que l'ambició objectiu inicial d'automatitzar completament el procés no ha estat possible donada la complexitat de les imatges, les eines de software disponibles i la dificultat per identificar "artificialment" les figures. Aquest fet impossibilita la pretesa automatització completa del procés on la intervenció de l'usuari es limitaria a seleccionar una imatge per a que l'aplicació li tornés les mesures de totes les partícules que la integren.

Pel que fa al programa, s'han exposat una sèrie de funcionalitats extres que s'hi poden implementar en un futur per millorar-lo i fer-lo més amigable per l'usuari.

Mentre s'estava duent a terme el treball, es va assistir juntament amb el director del projecte, a la jornada "Comercialització de software: limitacions i oportunitats" organitzada per la UPC el dia 13 novembre del 2017. El motiu de l'assistència a aquesta activitat és per si fos convenient la protecció intel·lectual del programa així com conèixer una mica més les vies de comercialització de programari.

Agraïments

El meu agraïment al meu tutor del projecte, el professor Jordi Bou Serra, per donar-me la possibilitat de realitzar aquest treball. També vull agrair-li el temps que m'ha dedicat i la seva disponibilitat total quan he tingut alguna mena de consulta.

És un treball que m'ha impulsat a adquirir coneixements en un àmbit que era totalment desconegut per mi i que m'ha aportat experiències positives a nivell d'aprenentatge personal.

Bibliografia

- [1] F. Escolano, O. Colomina, M.A.Cazorla. *Visión Artificial: Extracción de Características I*, 2006.
- [2] Hill Green. *Canny Edge Detection Tutorial*. 2002.
- [3] *Python Documentation*, [http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_setup/py_table_of_contents_setup/py_table_of_contents_setup.html] (Nov 2017)
- [4] *Python Documentation*, [<http://docs.python.org.ar/tutorial/3/classes.html>] (Nov 2017)
- [5] *Python Documentation*, [<https://docs.python.org/3/library/tk.html>] (Nov 2017)
- [6] *OpenCV Documentation*, [http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_gradients/py_gradients.html#gradients] (Nov 2017)
- [7] *OpenCV Documentation*, [https://docs.opencv.org/trunk/d3/dc0/group__imgproc__shape.html#ga8d26483c636be6b35c3ec6335798a47c] (Nov 2017)
- [8] *Scipy Documentation*, [<https://docs.scipy.org/doc/numpy/reference/generated/numpy.ndarray.html>] (Oct 2017)
- [9] UNIVERSIDAD DE JAÉN, DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA, TELECOMUNICACIÓN Y AUTOMÁTICA. ÁREA DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA. *Reducción del ruido en una imagen digital*. Jaén, 2005. [<http://www4.ujaen.es/~satorres/practicas/practica2.pdf>] (Set 2017)
- [10] José J. Grimaldos, *Tratamiento digital de imágenes*. Almería, 2006. [<http://www.grimaldos.es/cursos/imgdig/index.html>] (Set 2017)